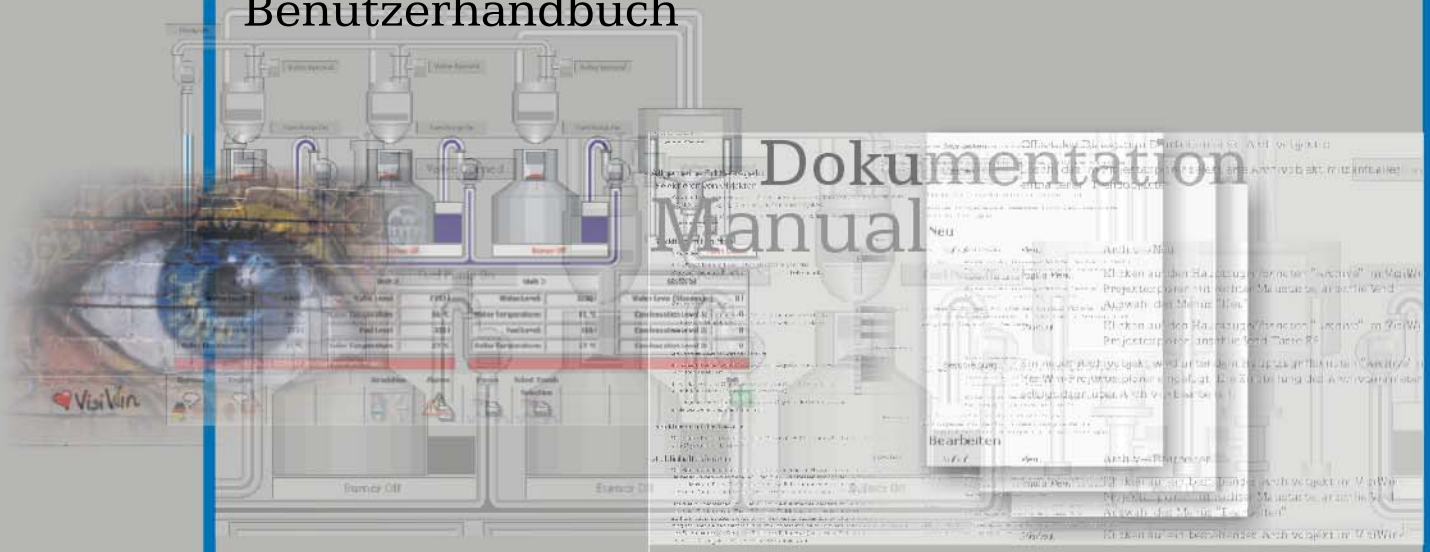
Automatisierungskonzepte

INOSOFT®

VisiWinNET Smart

Benutzerhandbuch



- VisiWin
  - VisiWinNET
    - Allgemeines
    - Klassenbibliothek
    - Systeme
    - **Werkzeuge**
    - Technische Infos
  - Inosoft OPCServer
    - Grundlagen und Hilfsmittel
    - Protokolle

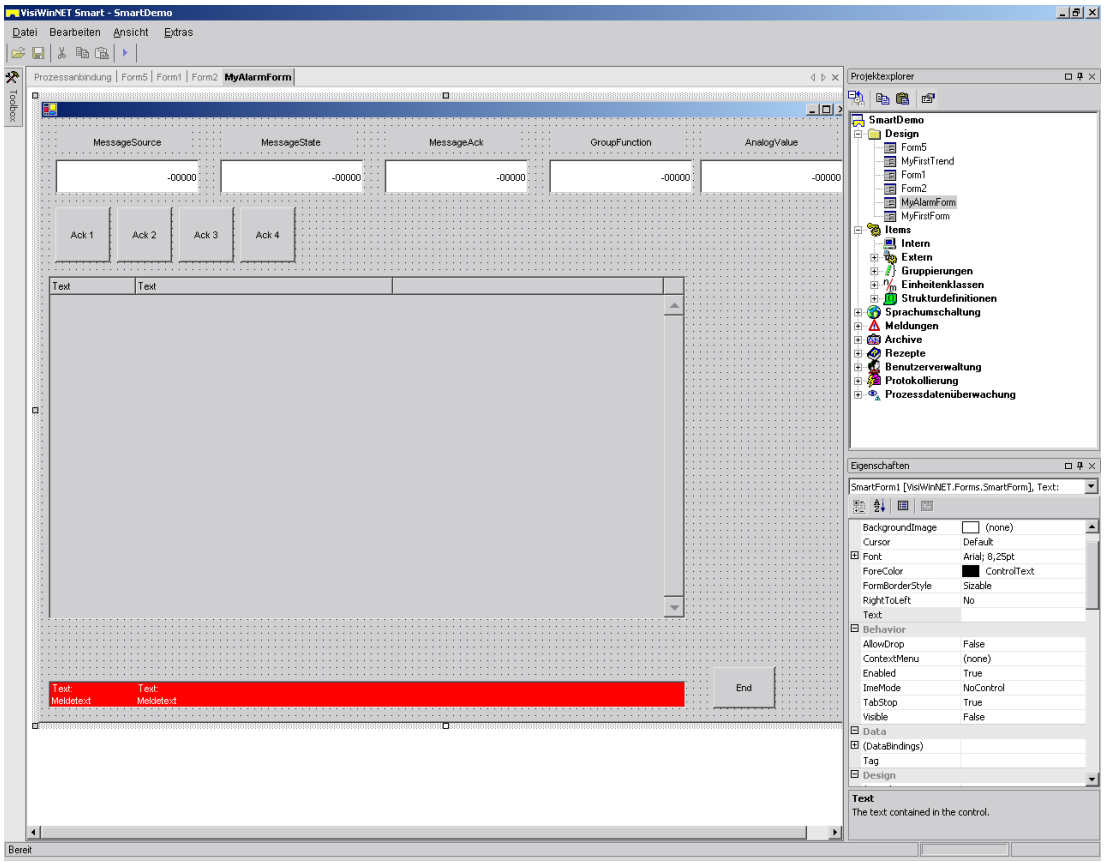
VisiWin

www.inosoft.com

Die Brücken zwischen Mensch und Maschine

VisiWinNET Smart

Benutzerhandbuch



Smart Entwicklungsumgebung

Ohne ausdrückliche schriftliche Erlaubnis der INOSOFT GmbH darf der Inhalt des Handbuches nicht anderweitig verwendet werden.
Den Inhalt des Handbuches haben wir auf Übereinstimmung mit der beschriebenen Software geprüft. Trotzdem können Abweichungen nicht ausgeschlossen werden. Aus diesem Grund können wir für eine völlige Übereinstimmung keine Gewähr geben. Der Inhalt des Handbuches wird regelmäßig auf Änderungen geprüft. Korrekturen sind in der nachfolgenden Ausgabe vorhanden. Verbessungsvorschläge werden dankbar angenommen.

Legende

Um auf besondere Einstellungen aufmerksam zu machen, werden in den INOSOFT-Dokumentationen folgende Symbole verwendet:

Achtung	Absätze mit diesem Zeichen sollten Sie besonders aufmerksam lesen - und beachten.
Hinweis	Wichtige Einstellung "additional informations".
Tipp	Viele Wege führen nach Rom, hier finden Sie eine Abkürzung.
In Arbeit	Funktionen in Vorbereitung, bzw. bereits implementiert, jedoch noch nicht für die Dokumentation aufbereitet.
Beispiel ausführen	Anweisungen, die in einem Beispiel auszuführen sind.
Beispiel beobachten	Ergebnisse, die durch die Ausführung der Beispielanweisungen zu beobachten sind.

© / ™ / ®
Windows®, Windows NT®, Windows 2000® , Windows XP®
sind eingetragene Warenzeichen der Firma Microsoft.
Weitere mit ® erwähnte Produktnamen sind Warenzeichen der betreffenden Herstellerfirmen.

INOSOFT GmbH

VisiWinNET Version: ab 5.00.0000

erstellt am 30.01.2006

Inhalt

1 Vorwort 1

2 Einleitung 2

2.1 Schnelleinstieg 3

2.1.1 Steuerungsanbindung mit OPC 4

2.1.2 Steuerungsanbindung mit VisiWin-Treibern 9

2.1.3 Darstellung und Manipulation von Prozesswerten in der Applikation 17

2.1.4 Grundregeln Applikationsaufbau 22

2.1.5 Sprachumschaltung 25

2.1.6 Meldesystem 33

2.1.7 Archivsystem 43

2.1.8 Rezepte 45

2.1.9 Benutzerverwaltung 51

3 Entwicklungsumgebung 58

3.1 Projektexplorer 61

3.1.1 Bedienung des VisiWinNET Projektexplorers 62

3.1.2 Schaltflächenbereich 63

3.1.3 Spezielle Kontextmenüs 71

3.2 Formulardesigner 72

3.3 Tabelleneditoren 74

3.4 Eigenschaftenfenster 75

3.5 Toolbox 77

3.6 Menüs 78

3.6.1 Datei 78

3.6.2 Bearbeiten 83

3.6.3 Ansicht 84

3.6.4 Zielgerät 86

3.6.5 Extras 88

3.7 Werkzeugleisten und Schaltflächen 89

3.7.1 Werkzeugleiste "Layout" 89

1 Vorwort

Zu diesem Handbuch

Dieses Handbuch richtet sich an den Visualisierer. Es zeigt die einzelnen Elemente der Entwicklungsumgebung "VisiWinNET Smart" und deren Verwendung. Meist enthält jedes der hier exemplarisch verwendeten Elemente eine eigene Referenz. Tiefergehende Informationen über einzelne Parametereinstellungen sind in den entsprechenden Referenzhandbüchern zu finden.

Das hier vorliegende Handbuch ist der erste Teil einer zweibändigen Serie zum VisiWinNET Smart. Weiterführende Informationen zur Erweiterung der Entwicklungsumgebung und Konzepte zur automatisierten Projekterstellung befinden sich im zweiten Band "Smart Programmierhandbuch".

Fragen und Probleme

Bei technischen Fragen und Problemen wenden Sie sich bitte an Ihre zuständige INOSOFT Vertretung oder an den Support der INOSOFT GmbH unter Tel.:+49 (5221) 16 66 02 bzw. E-Mail: Support@INOSOFT.com

Häufige Fragen und Probleme werden auch auf unserer Homepage unter www.inosoft.com behandelt.
Dort finden Sie auch einen Supportbereich, in dem Sie sich mit Fragen direkt an unsere Zentrale wenden können.

2 Einleitung

Herzlich Willkommen

Wir heißen Sie hiermit kurz Willkommen in der neuen und faszinierenden Welt des "VisiWinNET Smart".

Zur Sache: Sie haben die Aufgabe, eine Anwendung unter Windows zu gestalten. Diese Anwendung soll Daten aus einem laufenden Prozess, einer Maschine oder einer Anlage darstellen, vielleicht auch beeinflussen. Das ganze wird "Prozessvisualisierung" genannt und Sie sind mehr oder weniger mit der Thematik vertraut.

Es liegt in Ihrem Interesse, den Weg zur fertigen Applikation auf dem kürzesten Weg zu erreichen. Sie erhoffen sich durch den Kauf dieses Produktes also im wesentlichen einen unkomplizierten und schnellen Weg zum Ziel. Zu einfach darf das Produkt für Ihre Zwecke aber auch nicht sein, denn Sie wollen die Individualität und Vorteile Ihrer Maschine oder Anlage in der Visualisierung repräsentieren.

Wir bieten Ihnen mit "VisiWinNET Smart" ein einfaches und trotzdem flexibles Werkzeug, das die Offenheit der VisiWinNET-Produktlinie voll widerspiegelt: "VisiWinNET Smart" ist die Lösung für Techniker, SPS-Programmierer und Automatisierer, die sich nicht um die Hintergründe scheren oder scheren wollen. Für diese Anwendergruppe bildet "VisiWinNET Smart" eine abgeschlossene und einfach zu bedienende Entwicklungsumgebung zur Erstellung von Prozessvisualisierungen.

Trotzdem haben Sie mit "VisiWinNET Smart" keinen festgeschriebenen Funktionsumfang erworben. Das Erfolgskonzept verbirgt sich in der Offenheit und Integrationsfähigkeit der einzelnen Elemente. In Verbindung mit dem Standard-Entwicklungspaket von VisiWinNET prägen Sie oder Ihr Systemprogrammierer die individuellen Merkmale Ihrer Visualisierung. Selbst erstellte Steuerelemente oder Formularvorlagen lassen sich nahtlos in "VisiWinNET Smart" überführen und dort verwenden. Die für die Prozessanbindung nötigen Definitionen werden entweder direkt in Projekttemplates hinterlegt oder sind auf sehr einfache Weise über die Windows-Zwischenablage manuell oder automatisiert von einem Projekt ins nächste zu überführen. Durch diese Offenheit wird erreicht, dass sich "VisiWinNET Smart" selbst nahtlos in Ihre firmeneigene Entwicklungsphilosophie einpasst.

"VisiWinNET Smart" in der von uns ausgelieferten Form wird damit zum Angebot, aber nicht zur Pflicht.

Inhalt dieses Handbuches

Dieses Handbuch richtet sich an den Visualisierer. Es zeigt die einzelnen Elemente der Smart-Entwicklungsumgebung und deren Verwendung. Meist enthält jedes der hier exemplarisch verwendeten Elemente eine eigene Referenz. Tiefergehende Informationen über einzelne Parametereinstellungen sind in den entsprechenden Referenzhandbüchern zu finden.

Das hier vorliegende Handbuch ist der erste Teil einer zweibändigen Serie zum Smart-Produkt. Weiterführende Informationen zur Erweiterung der Smart-Entwicklungsumgebung und Konzepte zur automatisierten Projekterstellung befinden sich im zweiten Band "Smart Systementwicklung".

2.1 Schnelleinstieg

Das folgende Kapitel vermittelt als praktisches Beispiel einen Überblick über die Funktionsweise der Entwicklungsumgebung, des Laufzeitsystems und der einzelnen Systeme, die spezielle visualisierungstypische Aufgaben übernehmen.

Das folgende Testszenario ermöglicht den Aufbau einer voll funktionsfähigen Applikation. Die Gliederung des Szenarios entspricht der Einfachheit halber teilweise der logischen Aufteilung von VisiWinNET, teilweise auch praktischen Grundsätzen der Visualisierungsentwicklung:

Steuerungs-anbindung mit OPC-Servern oder VisiWin-Treibern

Herstellung der Kommunikation mit der SPS: VisiWinNET unterstützt zwei verschiedene Arten von Kommunikationskomponenten:

- OPC-Server sind als standardisierte Kommunikationskomponenten für mittlerweile fast jede Steuerung erhältlich und binden den Projekteur nicht an ein spezifisches Visualisierungsprodukt, da mittlerweile fast jedes Visualisierungspaket den OPC-Standard beherrscht.
- VisiWin-Treiber als Ergebnis langjähriger Entwicklung insbesondere auch für langjährige Kunden oder Umsteigern von älteren Produkten (VisiWin32/VisiWinStudio). Sie stehen als Alternative zu OPC-Servern weiterhin zur Verfügung, gegebenenfalls auch für Steuerungen, für die kein OPC-Server existiert und für die im Rahmen einer Sonderentwicklung ein Treiber erstellt wurde.

Das Beispiel zeigt beide Anbindungsarten anhand der beim Produkt beiliegenden Demonstrationskomponenten.

Die Anbindung an VisiWin-Treiber ist in Projekten für das Betriebssystem "Windows CE" nicht möglich.

Darstellung und Manipulation von Prozesswerten in der Applikation

Die Applikation wird mit Leben gefüllt. Erste, sich ändernde Prozesswerte werden auf unterschiedliche Arten dargestellt. Das Schreiben von Prozesswerten wird mithilfe spezieller Steuerelemente vorgenommen.

Grundregeln Applikationsaufbau	Hier wird gezeigt, wie eine Applikation modular aufgebaut wird. Die Unterteilung und der Wechsel zwischen den einzelnen Bildschirmansichten wird erklärt.
Sprachumschaltung	Die Schnittstelle zur Sprachumschaltung wird mit Daten ausgefüllt und in Betrieb genommen.
Meldesystem	Fehlerzustände ermitteln und darstellen.
Archivsystem	Sinnvolle Langzeitmessungen an Prozesswerten.
Rezepte	Parametersätze darstellen, optimieren und zur Maschine senden.
Protokollierung	Alles muss nachvollziehbar sein. Aufzeichnen von Benutzerinteraktionen und SPS-Werten, so dass eine Chronik der Maschine entsteht.
Benutzerverwaltung	Verschiedene Benutzer dürfen nur begrenzten Zugriff auf die Bestandteile der Applikation erlangen.

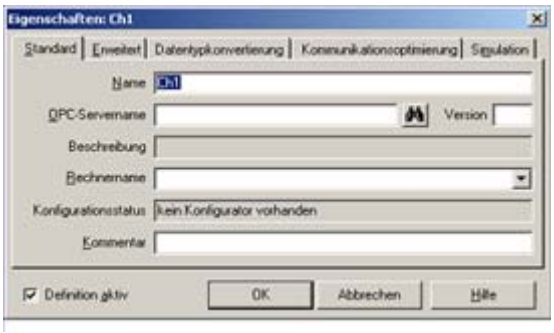
2.1.1 Steuerungsanbindung mit OPC

In diesem Kapitel wird das Anlegen eines neuen OPC-Servers und die Übernahme von OPC-Items in das VisiWinNET-Testprojekt beschrieben.

Ein wesentlicher Bestandteil einer Visualisierung unter VisiWinNET ist die Projektierung der Prozessanbindung. Hier wird festgelegt, welche Daten aus der SPS für die Visualisierung benötigt werden, mit welchen Kommunikationseinstellungen sie ausgetauscht werden und, wenn Bedarf besteht, wie diese Daten in der Visualisierung benannt werden. Die Prozessanbindung wird im Itemeditor vorgenommen. Mit dem Itemeditor werden Definitionen festgelegt, die bestimmen, welche Prozesswerte mit der SPS ausgetauscht werden sollen. Der Variablenkernel (zentrale Komponente zum Austausch von Prozesswerten zwischen SPS und Visualisierung) interpretiert diese Definitionen zur Laufzeit und stellt den entsprechenden Kontakt mit den Kommunikationskomponenten her. Der Itemeditor wird durch den Knoten "Items" repräsentiert.

Der Knoten "Items" im VisiWinNET Projektexplorer ist zu expandieren. Der Eintrag "neuer Kanal (OPC-Server)" im Kontextmenü des Knotens "Extern" (Kontextmenüs werden durch einen Klick mit der rechten Maustaste geöffnet) ist zu aktivieren.

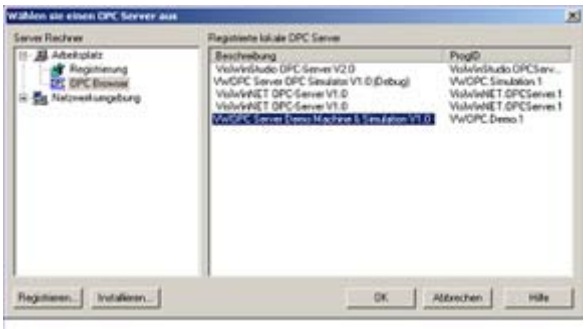
Es erscheint nun ein Fenster mit dem Titel "Eigenschaften für Ch1".



Dies ist der Dialog zur Festlegung der Parameter des neuen Kommunikationskanals. Kommunikationskanäle dienen zur Festlegung der Kommunikationskomponente und des Datenaustauschverhaltens. Eine Kommunikationskomponente ist ein SPS-spezifisches Treiberprogramm, das das Datenaustauschprotokoll mit der SPS unterstützt. VisiWinNET unterstützt zwei Kategorien von Kommunikationskomponenten: "OPC-Server" und "VisiWin-Treiber". Da die beiden Kategorien grundsätzlich verschieden Kommunikationsarten beinhalten, wird schon beim Anlegen eines neuen Kommunikationskanals unterschieden. Die Möglichkeiten von VisiWin-Treibern werden im folgenden Kapitel erläutert.

Die beteiligte Kommunikationskomponente wird im Eingabefeld "OPC-Servername" festgelegt. Dabei wird die sogenannte "ProgID" (Name, unter dem die Softwarekomponente im Windows-Betriebssystem registriert wird und als COM-Objekt ansprechbar ist) als Eingabe erwartet.

Der Dialog zur Auswahl eines OPC-Servers ist über die -Schaltfläche (rechts neben dem Eingabefeld) zu öffnen.



Der Dialog zur Auswahl eines OPC-Servers listet alle auf dem Rechner installierten OPC-Server auf. Bei der Installation von VisiWinNET wird unter anderem auch der für Schulungs- und Demonstrationszwecke erstellte "Demo-OPC-Server" in das System eingespielt. Dieser simuliert die SPS der verschiedenen Demoapplikationen des Entwicklungspakets.

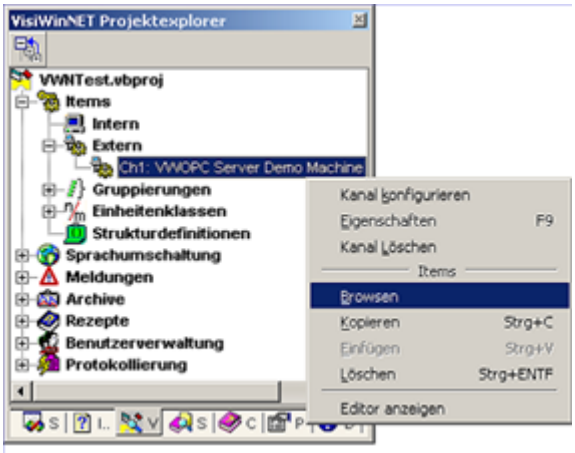
In der Auswahl ist der Eintrag "VWOPC Server Demo Machine & Simulation" zu wählen. Über die OK-Schaltfläche ist die Auswahl zu bestätigen.

Die ProgID des gewählten OPC-Servers wird in das Feld OPC-Servername übernommen. Damit ist die Wahl der Kommunikationskomponente getroffen. Über die Definition des Kommunikationskanals wird der Kontakt zur Kommunikationskomponente hergestellt. Im Folgenden sind noch die Festlegungen der auszutauschenden Prozesswerte vorzunehmen. Dies geschieht über die Definition von Items.

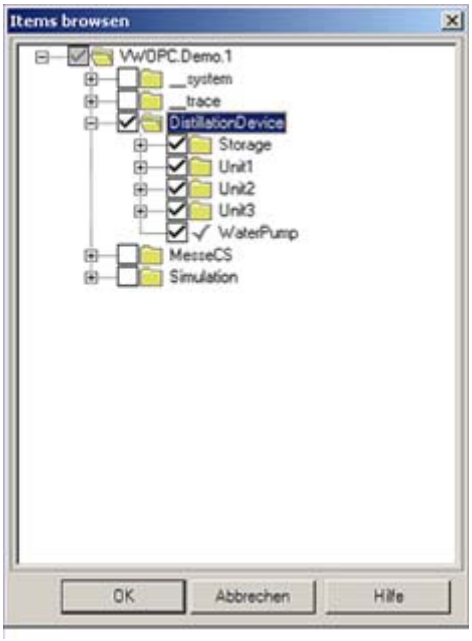
Die Eingaben im Dialog zur Festlegung der Parameter des Kommunikationskanals sind über die OK-Schaltfläche zu bestätigen.

Der Dialog wird geschlossen. Im Projektexplorer wird unter dem Zugriffsknoten "Extern" ein Knoten "Ch1: VWOPC Server Demo Machine & Simulation" eingefügt. Dieser repräsentiert den Kommunikationskanal. Das Kontextmenü des Knotens erlaubt weitere Funktionen, die auf diese Definition angewendet werden können.

Über das Kontextmenü des Kommunikationskanals ist die "Browsen"-Funktion zu aktivieren.



Es öffnet sich nun ein Fenster mit dem Titel "Items browsen".



Der Itembrowser zeigt alle im OPC-Server befindlichen Itemdefinitionen hierarchisch an. In der obersten Hierarchie befindet sich der Name des OPC-Servers, in der untersten Hierarchie werden die einzelnen Items dargestellt. Dazwischen befinden sich die sogenannten "Branches". Diese stellen kein Objekt dar, sondern zeigen nur die Strukturierung der Items in Namensräumen an:

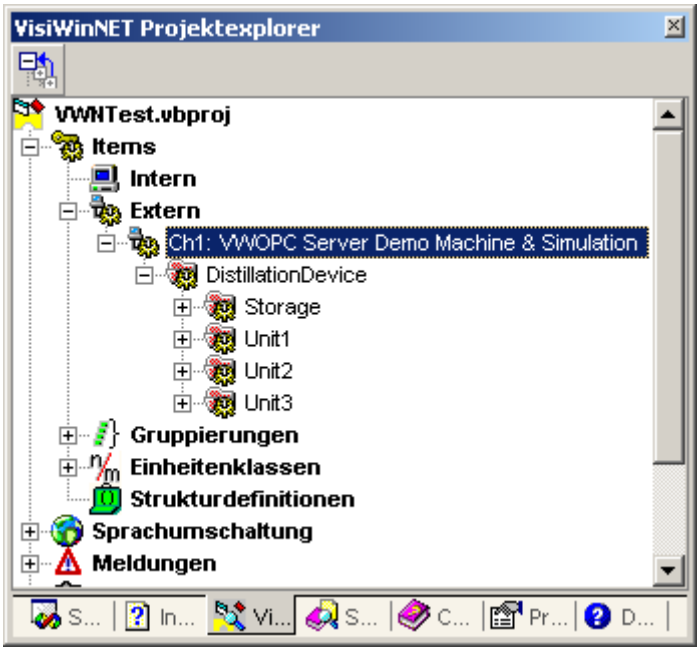
Itembezeichner werden durch ein Trennzeichen (ähnlich wie in der vollständigen Pfadangabe einer Datei das "\"-Zeichen) in Bezeichnerteile zerlegt. Items, die mit gleichen Bezeichnerteilen beginnen, werden unter einem gemeinsamen Branch zusammengefasst.

Wird ein Namensraum durch das Kontrollkästchen aktiviert, so werden dadurch auch alle darunter liegenden Namensräume und Itembezeichner mit aktiviert.

Durch die Auswahl eines Itembezeichners wird die entsprechende Itemdefinition in die VisiWinNET-Projektdatenbank übernommen.

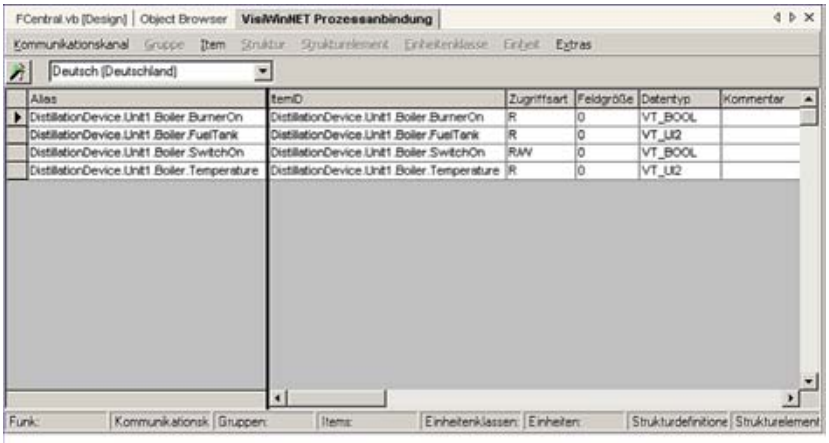
Der Knoten "Distillation Device" ist, wie in der Grafik dargestellt, auszuwählen. Die Auswahl ist über die OK-Taste zu bestätigen.

Die gewählten Items werden als Definitionen in die Projektdatenbank übernommen. Die Darstellung im Projektexplorer beschränkt sich auf die hierarchische Ansicht von Namensräumen.



Die Itemdefinitionen werden mitsamt ihrer Parameter im Tabelleneditor angezeigt.

Ein Doppelklick auf einen Namensraum öffnet den Tabelleneditor in der Entwicklungsumgebung (über dem Design-/Codefensterbereich).



Hier werden die Itemdefinitionen des gerade gewählten Namensraumes tabellarisch aufgelistet. Für das hier beschriebene Beispiel reicht es jedoch, die Itemdefinitionen aus dem OPC-Server zu übernehmen. Weitere Anpassungen sind nicht nötig. Der Tabelleneditor kann also wieder geschlossen werden.

2.1.2 Steuerungsanbindung mit VisiWin-Treibern

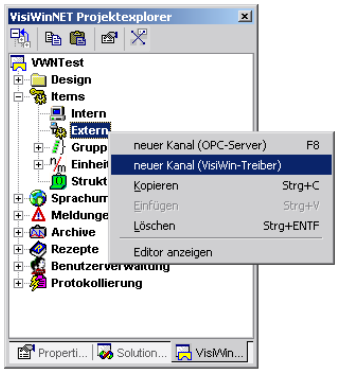
Die Anbindung an VisiWin-Treiber ist in Projekten für das Betriebssystem "Windows CE" nicht möglich.

In diesem Kapitel wird das Anlegen eines neuen VisiWin-Treibers und den entsprechenden Itemdefinitionen beschrieben. Außerdem wird die Funktion von Strukturen erläutert.

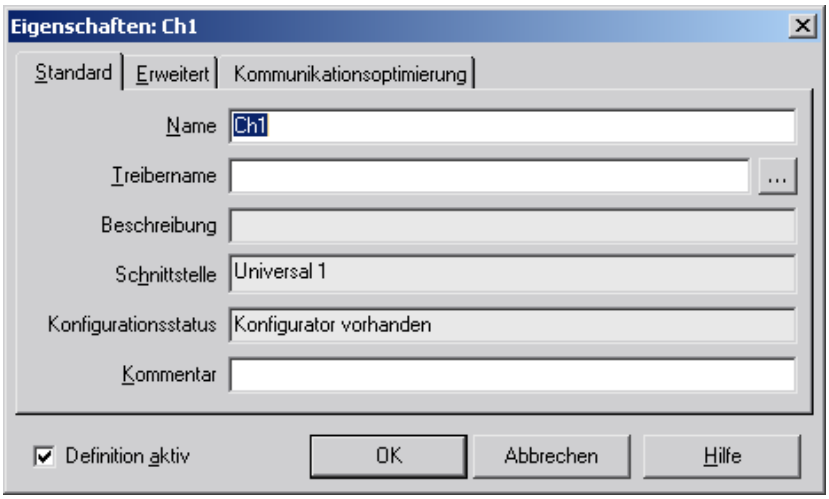
VisiWin-Treiber sind eine Alternative zu OPC-Servern. Sie werden üblicherweise verwendet, wenn kein OPC-Server für eine Steuerung vorliegt, wenn sehr spezielle Anforderungen über einen Kundenauftrag realisiert werden oder wenn bestehende Anbindungen aus älteren VisiWin-Visualisierungspaketen ohne Änderung übernommen werden sollen.

VisiWin-Treiber können nur in den Produktvarianten "Standard" und "Enterprise" verwendet werden. In "Compact" und "Embedded" fehlen die entsprechenden Funktionen.

Der Knoten "Items" im VisiWinNET Projektexplorer ist zu expandieren. Der Eintrag "neuer Kanal (VisiWin-Treiber)" im Kontextmenü des Knotens "Extern" (Kontextmenüs werden durch einen Klick mit der rechten Maustaste geöffnet) ist zu aktivieren.



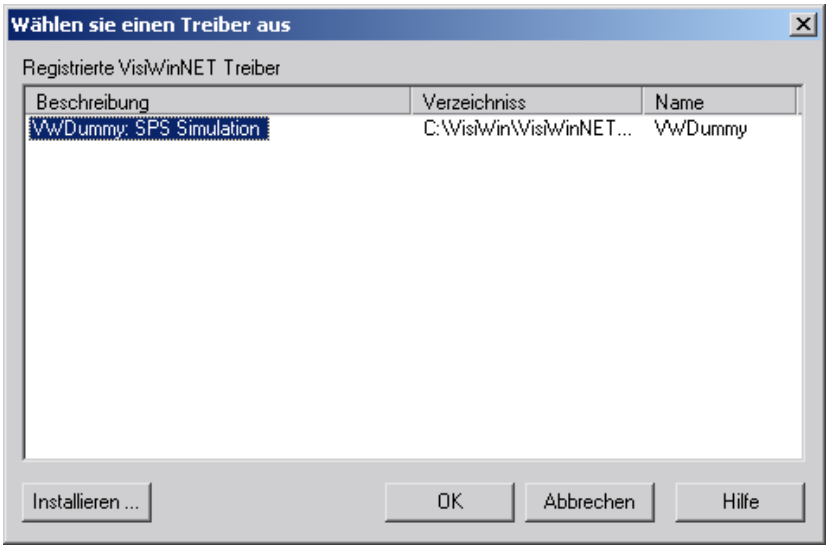
Es erscheint nun ein Fenster mit dem Titel "Eigenschaften für Ch1".



Dies ist der Dialog zur Festlegung der Parameter des neuen Kommunikationskanals. Kommunikationskanäle dienen zur Festlegung der Kommunikationskomponente und des Datenaustauschverhaltens. Eine Kommunikationskomponente ist ein SPS-spezifisches Treiberprogramm, das das Datenaustauschprotokoll mit der SPS unterstützt. VisiWinNET unterstützt zwei Kategorien von Kommunikationskomponenten: "OPC-Server" und "VisiWin-Treiber". Da die beiden Kategorien grundsätzlich verschieden Kommunikationsarten beinhalten, wird schon beim Anlegen eines neuen Kommunikationskanals unterschieden. Die Möglichkeiten von OPC-Servern werden im vorhergehenden Kapitel erläutert.

Die beteiligte Kommunikationskomponente wird im Eingabefeld "Treibername" festgelegt. Dabei wird der spezifische Bezeichner des Treibers aus der Registrierung (Treiber registrieren sich während der Installation).als Eingabe erwartet.

Der Dialog zur Auswahl eines VisiWin-Treibers ist über die Schaltfläche (rechts neben dem Eingabefeld) zu öffnen.



Der Dialog zur Auswahl eines VisiWin-Treibers listet alle auf dem Rechner installierten Treiber auf. Bei der Installation von VisiWinNET wird unter anderem auch der für Schulungs- und Demonstrationszwecke erstellte "VWDummy"-Treiber in das System eingespielt. Dieser simuliert einige Funktionen, die zum Aufbau von Testapplikationen verwendet werden können.

In der Auswahl ist der Eintrag "VWDummy SPS Simulation" zu wählen. Über die OK-Schaltfläche ist die Auswahl zu bestätigen.

Der Bezeichner des gewählten Treibers wird in das Feld "Treibername" übernommen. Damit ist die Wahl der Kommunikationskomponente getroffen. Über die Definition des Kommunikationskanals wird der Kontakt zur Kommunikationskomponente hergestellt. Im Folgenden sind noch die Festlegungen der auszutauschenden Prozesswerte vorzunehmen. Dies geschieht über die Definition von Items.

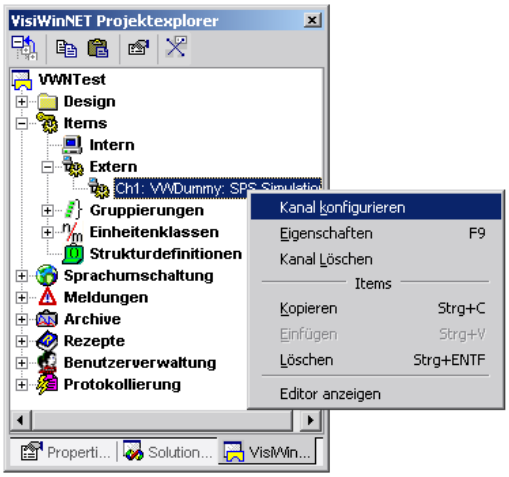
Die Eingaben im Dialog zur Festlegung der Parameter des Kommunikationskanals sind über die OK-Schaltfläche zu bestätigen.

Der Dialog wird geschlossen. Im Projektexplorer wird unter dem Zugriffsknoten "Extern" ein Knoten "Ch1: VWDummy SPS Simulation" eingefügt. Dieser repräsentiert den Kommunikationskanal. Das Kontextmenü des Knotens erlaubt weitere Funktionen, die auf diese Definition angewendet werden können.

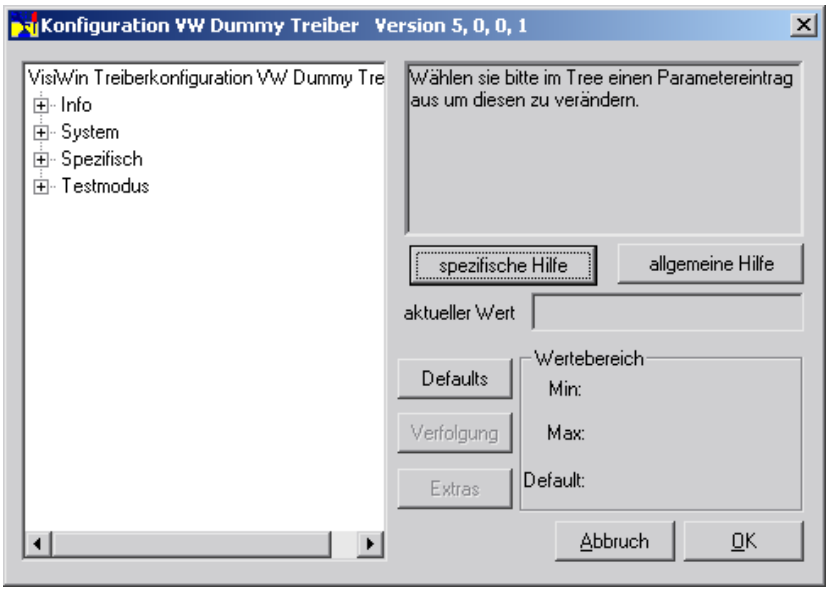
Itemdefinitionen in Treibern sind nicht automatisiert anzulegen. Sie müssen per Hand durch das Hinzufügen einzelner Datensätze im Tabelleneditor definiert werden.

Ein spezifisches Merkmal sind die Adressen, über die der Treiber Daten mit der SPS austauscht. Jedem Item muss eine Adressangabe und (über den Datentyp) eine Länge zugewiesen werden. Dazu kommt, dass Adressangaben je nach SPS-Typ symbolisch oder direkt zu verwenden sind. Verschiedene Treiber erwarten also unterschiedliche Adressangaben. Die Information, welcher Syntax die Angabe einer SPS-Adresse folgt, ist in der Treiberhilfe dokumentiert. Der schnellste Weg, diese Hilfe aufzurufen, geht über den Menüeintrag "Kanal konfigurieren". Hier erscheint ein treiberspezifischer Konfigurationsdialog, der generelle Einstellungen zum Treiber zulässt:

Über das Kontextmenü des Kommunikationskanals ist die "Kanal konfigurieren"-Funktion zu aktivieren.

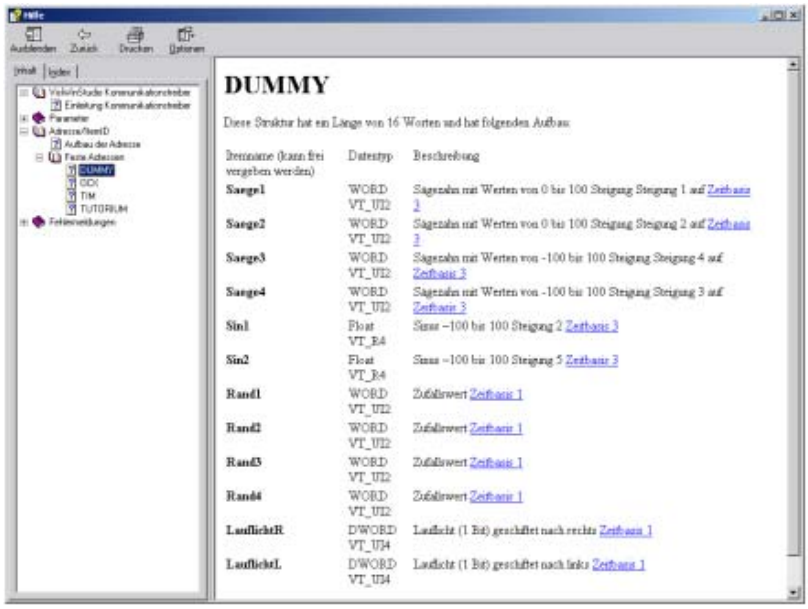


Es öffnet sich nun ein Fenster mit dem Titel "Konfiguration VWDummy Treiber...".



Über die Schaltfläche "spezifische Hilfe" ist die Hilfe zum Treiber zu öffnen.

Alle Treiberhilfen enthalten einen Abschnitt über den Aufbau der spezifischen Adressangaben (Adresse/ItemID)



Was bei einer realen SPS erst durch die Programmierung festgelegt wird, ist im Dummy-Treibers bereits fest verdrahtet. Über die hier angegeben Adressen können simulierte Variablenwerte abgerufen werden. Der Abschnitt "Adresse/ItemID→feste Adressen→Dummy" beschreibt die Datenstruktur der "Dummy"-Adresse.

Wieso "Datenstruktur der Dummy"-Adresse? Sind "Saege1", "Saege2" usw. denn keine Adressen?

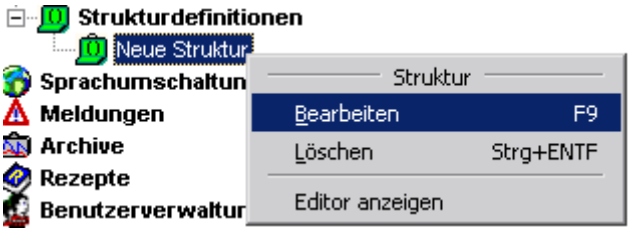
Nein, sind sie nicht. Treiber haben grundsätzlich die Möglichkeit, mit der Steuerung Daten einzeln oder blockweise auszutauschen. Die Definition eines Items als Grunddatentyp "VT_..." ermöglicht der Zugriff auf ein einzelnes Item. Die Definition eines Items vom Datentyp einer Struktur ist ein blockweiser Datenaustausch. Innerhalb des Blockes liegen die einzelnen Variablen, deren Länge durch den Datentyp bestimmt wird. Eine Variable vom Datentyp VT_I4 z.B. hat eine feste Länge von vier zu übertragende Bytes. Die Übertragung eines Blockes erfordert nur eine steuerungsspezifische Startadresse und die Länge des Datenblockes, die sich im Fall von Strukturen über die Länge der einzelnen Elemente berechnen lässt.

Eine Struktur bildet sich aus einzelnen Elementen, die wiederum Strukturen oder Grunddatentypen sein können. Strukturen haben den Vorteil, dass mit einem minimalen Protokoll-Overhead viele Daten gelesen oder geschrieben werden können. Dies ist insbesondere dann zwingend erforderlich, wenn die SPS-Schnittstelle langsam ist (z.B. seriell 9600 Baud). Über Strukturen lässt sich also die Kommunikation optimieren. Für die Übertragung einzelner SPS-Variablen muss nicht jedes mal ein Protokollrahmen gesendet werden. Über einen einzigen Protokollrahmen kann bei entsprechender Projektierung beispielsweise ein ganzer Datenbaustein transferiert werden. Die Auflösung in die einzelnen Elemente oder Variablen übernimmt das Laufzeitsystem in der Applikation.

Strukturen werden in VisiWinNET definiert.

Über das Kontextmenü des Knotens "Strukturdefinitionen" ist eine neue Struktur zu definieren.

Die als "neue Struktur" eingetragene Definition ist über das Kontextmenü als Dialog zu öffnen.



Dort ist der Name in sDummy zu ändern.

Über einen Doppelklick auf den Knoten "sDummy" ist der Tabelleneditor zu öffnen. Hier sind 12 neue Strukturelemente anzulegen (Klick auf die leere Tabelle anschließend 12 x "F8" drücken.

Die Parameter der neuen Strukturelemente sind wie in der Hilfe angegeben zu ändern:

Name	Datentyp
Saege1	**VT_UI2**
Saege2	**VT_UI2**
Saege3	**VT_UI2**
Saege4	**VT_UI2**
Sin1	**VT_R4**
Sin2	**VT_R4**
Rand1	**VT_UI2**
Rand2	**VT_UI2**
Rand3	**VT_UI2**
Rand4	**VT_UI2**
LauflichtR	**VT_UI4**
LauflichtL	**VT_UI4**

Abschließend ist zu überprüfen, ob der Parameter "Ordnungsnummer" fortlaufend in oben genannter Reihenfolge aufsteigt.

Die hier aufgebaute Struktur beschreibt den Aufbau eines Datenblockes. Die Struktur selbst ist aber noch kein Baustein, der selbst Daten austauscht. Er ist nur eine Typendeklaration. Diese Deklaration kann in den Itemdefinitionen als Datentyp verwendet werden.

Im Projektexplorer ist zum Knoten "Ch1: VWDummy..." zu wechseln.

Die Ansicht im Tabelleneditor zeigt die Itemdefinitionen des Kanals. Da noch keine Items definiert wurden, ist die Tabelle leer.

Auf den leeren Tabelleneditor ist zu klicken und über die Taste "F8" ein neues Item anzulegen. In diesem neuen Item sind folgende Werte zu ändern:

Parameter	Wert
Name	DummyItem
Adresse	Dummy
Datentyp	sDummy

Der Name des Items ist grundsätzlich wahlfrei. Ein zweckgebundener Name, aus dem die Funktion eines Items hervorgeht, erleichtert in den folgenden Entwicklungsschritten die Projektierung enorm.

Die angegebene Adresse bezieht sich auf die in der Hilfe beschriebene Funktion des Treibers.

Als Datentyp wird in der Auswahlliste die definierte Struktur angeboten. Die Definition eines Items vom Datentyp einer Struktur bewirkt zur Laufzeit folgenden Ablauf:

Während der Initialisierungsphase lädt der Variablenkern die Itemdefinitionen aus der Projektdatenbank und baut ein entsprechendes Speicherabbild auf. Dazu ermittelt er den Aufbau und die Länge der als Datentypen angegebenen Strukturen.

Bei der Anforderung von Werten aus der Applikation übersendet der Variablenkern die Startadresse und die Länge des Items. Im oben genannten Beispiel ergibt sich die Länge des angeforderten Datenblockes mit 32 Byte.

Der Treiber liest die angegebene Anzahl von Bytes ab der übergebenen Startadresse mit einem Protokollzugriff und gibt den Byteblock an den Variablenkern zurück.

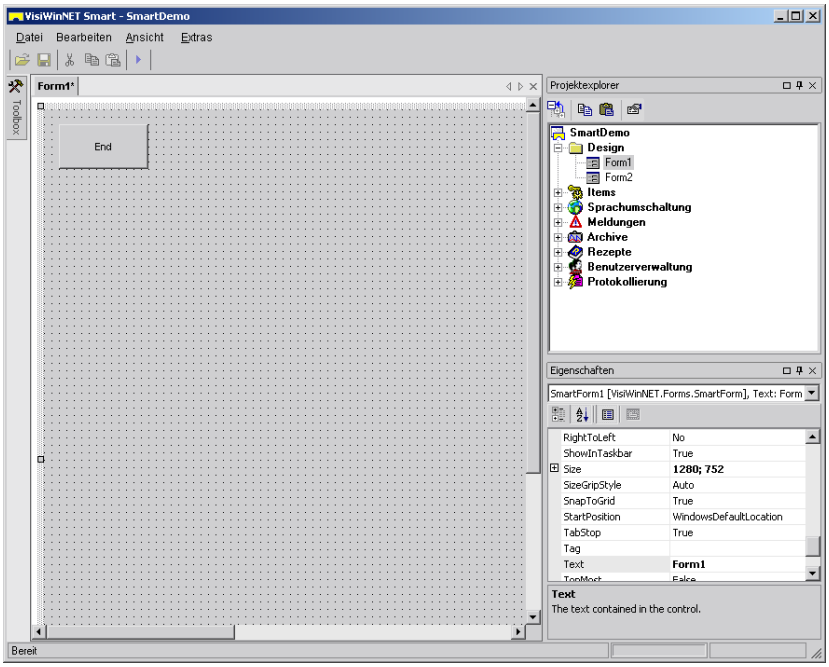
Der Variablenkern ordnet jetzt die einzelnen Bytes intern zu den entsprechenden Variablen im Speicherabbild zu.

2.1.3 Darstellung und Manipulation von Prozesswerten in der Applikation

Dieses Kapitel beschreibt den allgemeinen Umgang mit Steuerelementen und stellt den Zusammenhang zu den Itemdefinitionen aus der Projektdatenbank dar.

Nach dem Anlegen eines neuen Projektes ist mindestens ein Formular bereits angelegt. Ein Formular repräsentiert eine Bildschirmansicht oder einen Dialog.

Um die Designansicht des Formulars zu öffnen, ist der Knoten "Design" zu expandieren. Darunter werden die Formulare des Projektes aufgelistet. Ein Doppelklick auf einen Formularnamen öffnet den Designer.

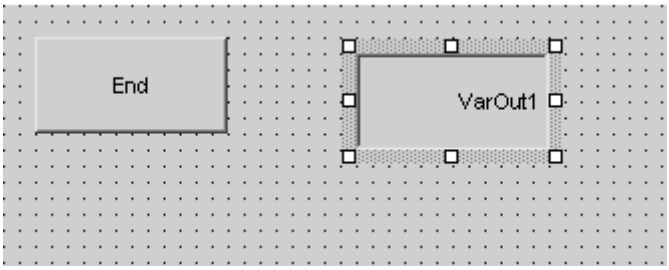


Der Designer stellt eine Bildschirmansicht dar. Das Aussehen der Bildschirmansicht wird über das Platzieren und Parametrieren von Steuerelementen bestimmt. Steuerelemente werden in der Toolbox ausgewählt. In der Grundeinstellung wird das Toolboxfenster oben links als Karteireiter dargestellt.

Die Toolbox ist über den entsprechenden Karteireiter zu expandieren. Der Steuerelementetyp "VarOut" ist in der Toolbox zu markieren.

Anschließend ist auf dem Formular die Stelle durch einen Klick mit der linken Maustaste zu wählen, wo das Steuerelement zu platzieren ist. Wird die Maustaste sofort wieder losgelassen, wird das Steuerelement an dieser Stelle mit einer vorgegebenen Größe platziert. Wird hingegen bei gehaltener Maustaste der Mauszeiger bewegt, bildet die Bewegung auf dem Bildschirm einen Rahmen. Das Loslassen der Maustaste bewirkt, dass das Steuerelement mit Größe und Position des dargestellten Rahmens auf dem Formular erzeugt wird.

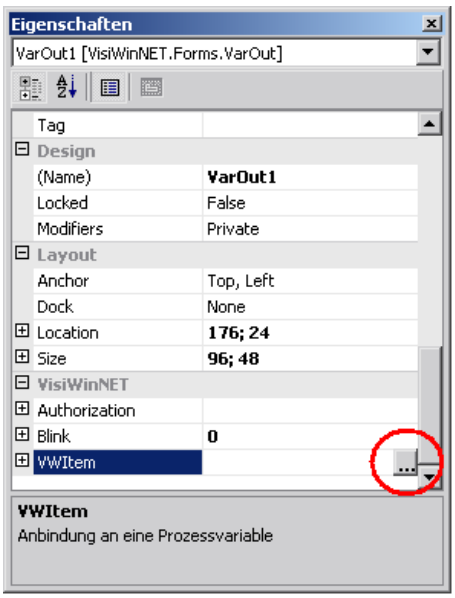
Nach dem platzieren erscheint das 'VarOut'-Steuerelement auf dem Formular



Der gewählte Steuerelementetyp dient zur numerischen oder alphanumerischen Anzeige von Variablenwerten. Die wesentliche Festlegung, welcher Prozesswert anzuzeigen ist, wird über die Eigenschaft "VWItem" vorgenommen.

Das Steuerelement ist mit einem Mausklick zu markieren, so dass wie oben gezeigt, der Selektionsrahmen sichtbar wird.

Im Eigenschaftenfenster ist über die Bildlaufleiste so weit zu blättern, bis die Eigenschaft "VWItem" sichtbar wird. Das leere Wertefeld der Eigenschaft ist über einen Mausklick zu markieren.

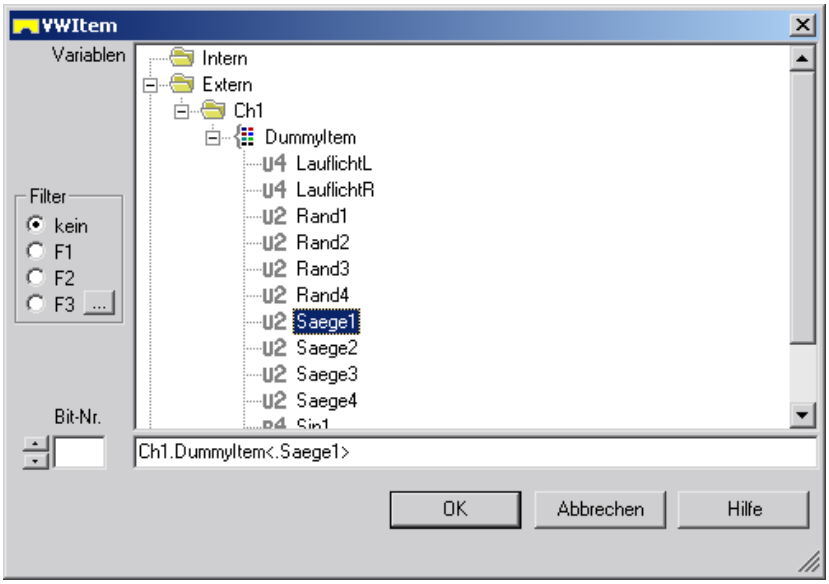


Im Wertefeld der Eigenschaft erscheint eine kleine Schaltfläche, über die ein Dialog zur Auswahl einer Variablen geöffnet werden kann.

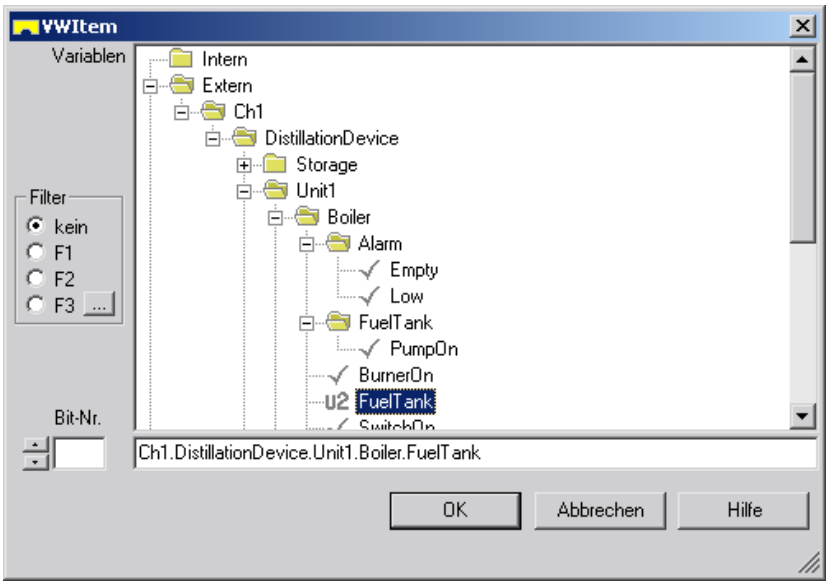
Die oben gezeigte Auswahlschaltfläche ist durch einen Mausklick zu bedienen.

Die vorhergehenden Kapitel beschreiben die Anbindung mit einem OPC-Server und einem VisiWin-Treiber. Je nachdem, welche Anbindung gewählt wurde, zeigt der Dialog folgenden Inhalt

Für VisiWin-Treiber...



...oder für OPC-Server



Der Dialog zeigt die zur Verfügung stehenden Prozessvariablen, die durch Browsen oder selbst Definieren in der Projektdatenbank gespeichert wurden. Im wesentlichen folgt der Aufbau der hier dargestellten Hierarchie dem Aufbau im Projektexplorer.

- "Intern" kennzeichnet Variablen, die keinem Kommunikationskanal zugeordnet sind, die also kein reales Abbild in einer Steuerung haben. Diese Variablen können trotzdem innerhalb einer Visualisierung benötigt werden, um z.B. variablengesteuerte Vorgänge auszulösen oder einfach als Puffer für Werte aus der Visualisierung.

"Extern" beinhaltet die Variablen, die ein SPS-Abbild haben. Unter extern werden die Kommunikationskanäle aufgelistet. Darunter kommen die Namensräume, die auch schon im Browsen verwendet wurden. Es folgen die Itemdefinitionen. Gegebenenfalls werden bei Treibern darunter noch die Strukturelemente zur Verfügung gestellt.

Die in den oben dargestellten Bildern selektierten Variablen sind auszuwählen. Über die OK-Taste ist die Auswahl zu bestätigen.

Durch die Festlegung der 'VWItem'-Eigenschaft ändert sich die Darstellung im Steuerelement "VarOut1". Bei der Festlegung des darzustellenden Items wird der Datentyp interpretiert. Die Anzeige in den Steuerelementen springt auf einen Wert, der die Darstellung zur Laufzeit simuliert. Damit ist es möglich, schon zur Entwicklungszeit zu überprüfen, ob das Steuerelement den Wert überhaupt vollständig anzeigen kann oder ob ggf. die Steuerelementeabmessungen geändert werden müssen.

Ein abschließender Test zeigt die Funktionsfähigkeit der Anbindung.

Der Testlauf eines Standard-Projektes gestaltet sich einfach:

In der Werkzeugleiste ist die -Schaltfläche zu drücken.

Ein Compact-Projekt erfordert eine Verbindung zum Zielsystem. Die Herstellung dieser Verbindung wird im Kapitel "Verbindung herstellen" (Kap. 3.6.4.1) beschrieben.

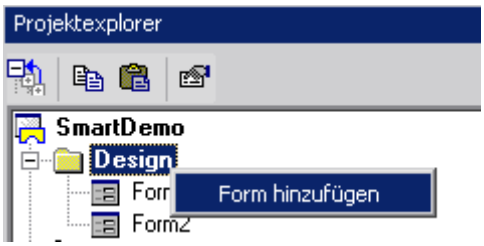
Das Projekt startet. Die nun angezeigte Ansicht entspricht der Oberfläche, die dem Visualisierungsbenutzer angezeigt wird. Ist die Funktion erfolgreich, sollte in der Darstellung nun ein sich ändernder Wert angezeigt werden.

VisiWinNET stellt eine Reihe weiterer Steuerelemente zur Verfügung, die die Anzeige oder Eingabe grafisch oder alphanumerisch ermöglichen. All diesen Steuerelementen gemeinsam ist die 'VWItem'-Eigenschaft, die die darzustellende oder zu schreibende Prozessvariable auswählt. Das Verfahren ist also bei allen Steuerelemente gleich oder zumindest ähnlich. Um einen Überblick über die Funktionen eines Steuerelementes zu gewinnen, kann es auf ein Formular gezogen, dort selektiert und über die Taste "F1" die Hilfeseite aufgerufen werden.

2.1.4 Grundregeln Applikationsaufbau

Selten wird eine Visualisierung mit nur einer Bildschirmansicht auskommen. Üblich und nützlich ist die Unterteilung in unterschiedliche Bildschirmansichten nach Funktion oder Logik der Anlage. Ausgehend von einer zentralen Übersichtsseite kann der Benutzer dann über Schaltflächen in die ihn betreffenden Applikationsteile wechseln und dort seine Eingaben oder Beobachtungen vornehmen. Zusätzlich zu beachten dabei ist die personenbezogene Freigabe von Funktionen. Allgemein üblich ist hier z.B. der Zugriff auf eine Service-Seite, die nicht vom Bedienpersonal, aber von einem Techniker aufgerufen werden darf.

In der Praxis werden die unterschiedlichen Bildschirmansichten durch Formulare gestaltet. Formulare werden im Projektexplorer über den Knoten "Design" hinzugefügt.



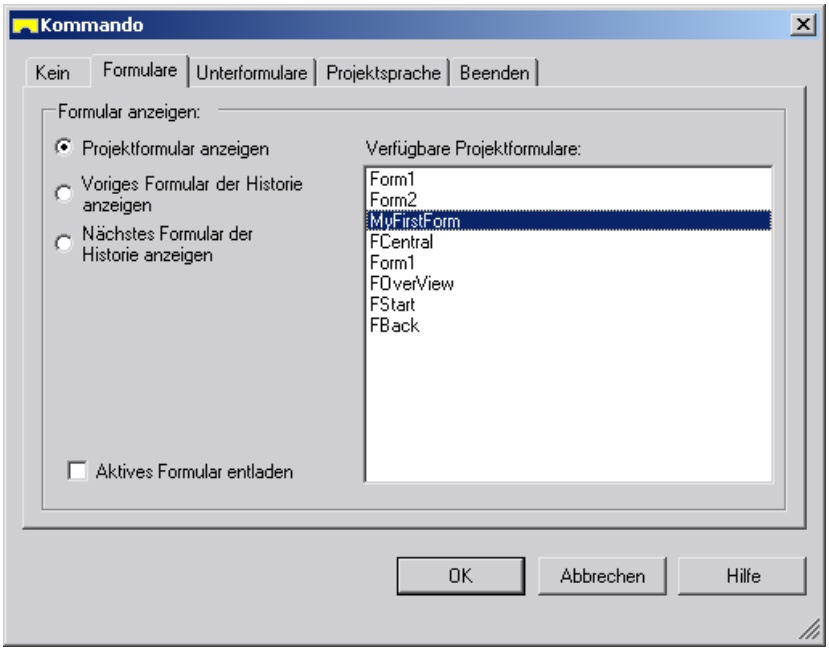
Ein neues Formular ist in ein Projekt einzufügen. Dazu ist der Knoten "Design" im Projektexplorer zu markieren und mit der rechten Maustaste das Kontextmenü aufzurufen. Im folgenden Dialog ist der Name "MyFirstForm" für das Formular anzugeben.

Ein neuer Knoten mit dem Namen "MyFirstForm" wird unter "Design" eingefügt. Ein entsprechender Formulardesigner wird geöffnet. Auf diesem Designer können wieder Steuerelemente platziert und parametriert werden.

Zu klären wäre jetzt noch die Frage, wie das neue Formular zur Anzeige gebracht wird.

Dazu ist zum ersten Formular der Anwendung zu wechseln. Auf diesem Formular ist ein Steuerelement vom Typ "CommandButton" zu platzieren.

Der Eigenschaftendialog der Eigenschaft "Command" des "CommandButton"-Steuerelementes ist zu öffnen. Auf der Karteikarte "Formulare" ist der Eintrag "MyFirstForm" in der Formularliste zu selektieren. Die Auswahl ist über die OK-Schaltfläche zu bestätigen.



Die hier festgelegte Funktion des "CommandButtons" bewirkt, dass zur Laufzeit das Formular "MyFirstForm" angezeigt wird.

Der Test über die -Schaltfläche zeigt, dass das Formular angezeigt wird, allerdings mit einer Titelzeile und den üblichen Funktionsschaltflächen darin.

Die gebräuchlichste Bildschirmansicht einer Visualisierungsapplikation ist ein Vollbild ohne Titelzeile.

Nach dem Beenden der Applikation ist das Formular "MyFirstForm" als Formulardesigner wieder sichtbar zu schalten.

Im Eigenschaftenfenster sind folgende Eigenschaften zu ändern:

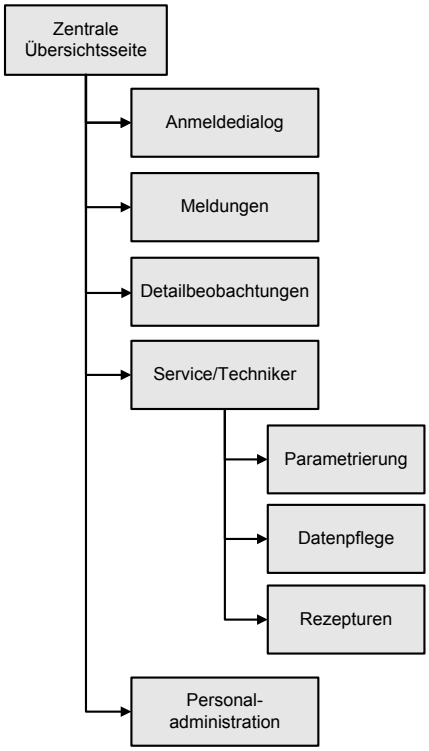
Eigenschaft	Wert
FormBorderStyle	None
WindowState	maximized

Da das Formular mit den hier eingestellten Eigenschaften jetzt keine Titelzeile mehr hat, kann es nicht ohne Hilfe einer Schaltfläche geschlossen werden. Auch kann es nicht mehr beiseite gerückt werden, da es maximiert den gesamten Bildschirminhalt füllt.

Auf dem Formular ist ein "CommandButton" zu platzieren. Die Eigenschaft "Command" ist so zu parametrieren, dass es wieder das erste Formular der Applikation aufruft.

Der Testlauf zeigt, dass es nun möglich ist, zwischen den beiden Formularen zu wechseln. Über den "CommandButton" sind also die einzelnen Seiten untereinander aufrufbar.

Eine wesentliche Überlegung beim Aufbau einer Visualisierung ist die Aufteilung der Bildschirmansichten. Ein typisches Beispiel soll hier gezeigt werden:



Jeder der hier dargestellten Funktionsblöcke soll durch eine oder mehrere Bildschirmansichten dargestellt werden. Eine so gezeichnete Hierarchie stellt die Grundzüge dar, wie die Seiten untereinander zu verknüpfen sind, welche Seiten also wo aufgerufen werden müssen. Wenn z.B. aus der zentralen Übersicht auf die einzelnen darunter liegenden Funktionsblöcke zugegriffen werden soll, sind mindestens 5 Schaltflächen nötig. Der Zugriff auf die Daten von Service/Techniker muss nicht auf der zentralen Übersichtsseite liegen, wenn eine Bildschirmansicht den Zugriff auf die Blöcke "Parametrierung", "Datenpflege" und "Rezepturen" enthält. Geschmackssache ist, ob parallel zueinanderliegende Bildschirmseiten untereinander verknüpft werden müssen. Eine Ausnahme hierbei ist jedoch mit Sicherheit eine Meldeansichtenseite, die auf jeden Fall schnell aus allen Teilen der Applikation erreichbar sein sollte.

2.1.5 Sprachumschaltung

Die Sprachumschaltung ermöglicht den Wechsel sämtlicher in der Applikation verwendeter Texte in Abhängigkeit von der gewählten Sprache. Sie wird verwendet um:

- Die Auslieferung einer international verwendbaren Applikation zu ermöglichen. Hier würde genügen, die anzuzeigende Sprache beim Projektstart oder auf einer Serviceseite zu konfigurieren.
- Internationalem Bedienpersonal die Bedienung zu ermöglichen. Denkbar ist hier, dass beispielsweise beim Schichtwechsel die Sprache umzustellen ist, da sich die Nationalität des Personal ändert.

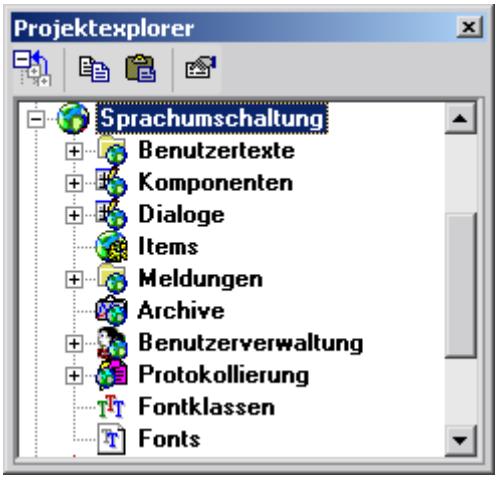
Beides ist quasi auf Knopfdruck möglich. Auch die Eingabe von nichtwestlichen Schriftzeichen ist über Unicode-Zeichensätze erlaubt.

Neben den eigentlichen Texten, die in der Applikation verwendet werden, sind weitere Besonderheiten zu beachten:

- In Abhängigkeit von der gewählten Sprache müssen Einheiten (z.B. Temperatur °C/°F) umgerechnet und umgestellt werden
- In Abhängigkeit von der gewählten Sprache müssen Datums- und Zeitformate im landesüblichen Format dargestellt werden.
- Gegebenenfalls ist die Größe von Schriftarten in unterschiedlichen Zeichensätzen anders normiert, so dass beim Wechsel der Sprache eine Anpassung der Schriftart nötig wird.

Die Angabe der Texte erfolgt in der Sprachumschaltung von VisiWinNET. Die Anbindung an die Steuerelemente erfolgt über verschiedene Eigenschaften.

Der Knoten "Sprachumschaltung" im Projektexplorer ist zu expandieren.



Die Funktionen der Sprachumschaltung ist in verschiedene Bereich gegliedert:

- "Benutzertexte" sind die Texte, die der Projekteur anlegen kann, um sie in der Applikation anzuzeigen.
- "Komponenten" und "Dialoge" enthalten die Texte, die von den Steuerelementen und Dialogen des VisiWin-Pakets verwendet und erwartet werden. Hier können keine Texte angelegt werden, die Texte können sprachlich erweitert oder applikationsbezogen geändert werden.
- in den Knoten "Items", "Meldungen", "Archive", Benutzerverwaltung" und "Protokollierung" können keine Texte angelegt werden. Das wird von den Editoren der entsprechenden Systeme erledigt. Wird beispielsweise eine Meldung im Meldesystem angelegt, erscheint der sprachumschaltbare Text der Meldung unter dem hier angezeigten Knoten. Dies entspricht dem zentralen Verwaltungsgedanken der Sprachumschaltung.
- "Fontklassen" und "Fonts" definieren die Funktion der Schriftartenumschaltung

Die Einheitenumschaltung ist im Itemsystem enthalten. Sie ist konzeptionell von der Sprachumschaltung getrennt worden, da die zu definierenden Informationen als Hauptmerkmal Umrechnungswerte enthalten.

Der Knoten "Benutzertexte" ist zu markieren. Über das Kontextmenü ist eine neue Textgruppe anzulegen. Die Textgruppe ist umzubenennen in "MyFirstForm".

Sprachumschaltbare Texte können in einzelnen Gruppen angelegt und verwaltet werden. Dies hat folgende Vorteile:

- Über den Namen der Textgruppe kann angegeben werden, wo die Texte verwendet werden. Über Textgruppen kann also eine Gliederung erfolgen.
- Die Texte, die angelegt und später vielleicht einmal geändert werden sollen, können leichter wiedergefunden werden.
- Wird zu jedem Formular eine Textgruppe angelegt, erleichtert dies später die Arbeit. Wenn das Formular in ein anderes Projekt übernommen wird, muss nur die entsprechende Textgruppe in das Projekt kopiert werden. Dann ist das Formular, das die Texte verwendet, wieder funktionsfähig.

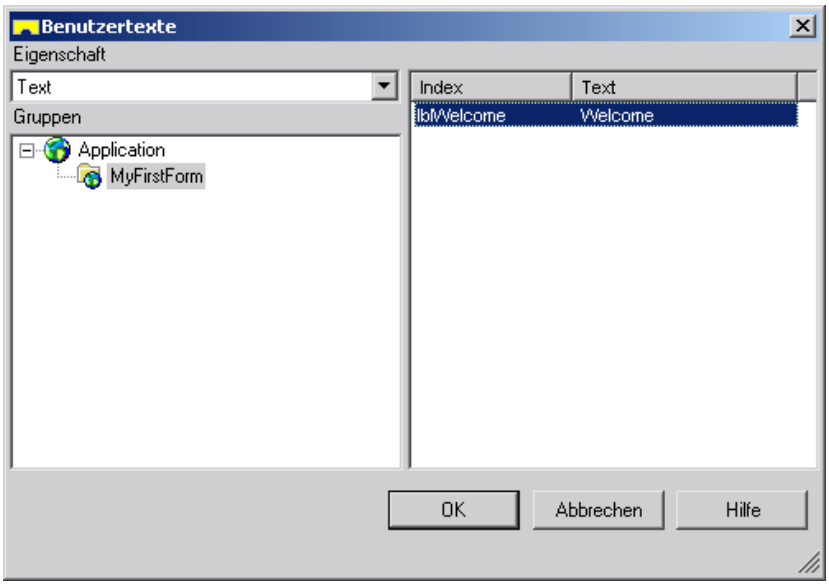
Der Tabelleneditor ist durch einen Doppelklick auf den neuen Knoten "MyFirstForm" zu öffnen. Der Tabelleneditor ist durch einen Klick zu markieren. Anschließend ist über die Taste "F8" ein neuer Text anzulegen. Der Text ist folgendermaßen anzupassen:

Parameter	Wert
Name	lblWelcome
Deutsch (Deutschland) (1031)	Willkommen
Englisch (USA) (1033)	Welcome

Über den entsprechenden Knoten unter Gruppe "Design" im Projektexplorer ist das Formular "MyFirstForm" zu öffnen. Hier sind ein 'Label'- und zwei 'CommandButton'-Steuerelemente zu platzieren.

Das 'Label'-Steuerelement ist zu markieren. Über die -Schaltfläche der 'LocalizedText'-Eigenschaft ist der Dialog zur Auswahl eines sprachumschaltbaren Textes zu öffnen.

Hier werden alle im Projekt angelegten Benutzertexte zur Auswahl angeboten.



In der Textgruppenansicht (links) ist zur Textgruppe 'MyFirstForm' zu navigieren. In der Textauswahlansicht (rechts) ist der Text "lblWelcome" zu markieren. Die Auswahl ist über "OK" zu bestätigen.

Über die Auswahl wird festgelegt, welcher Text vom Steuerelement zur Laufzeit angezeigt werden soll. Nach der Übernahmen wird im Steuerelement der Text in der aktiven Sprache angezeigt.

Die beiden 'CommandButton'-Steuerelemente sollen dazu dienen, die Sprache zur Laufzeit umzuschalten.

Das erste 'CommandButton'-Steuerelement ist zu markieren. Über die ...-Schaltfläche der 'Command'-Eigenschaft ist der Dialog zur Funktionsauswahl zu öffnen. Auf der Karteikarte "Projektsprache" ist der Eintrag "Deutsch (Deutschland)" auszuwählen. Die Auswahl ist über die "OK"-Schaltfläche zu bestätigen.

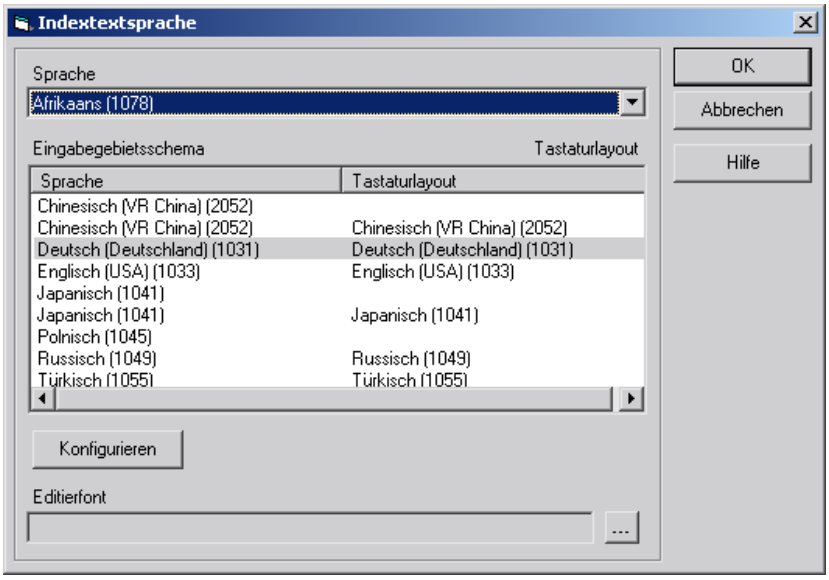
Für das zweite 'CommandButton'-Steuerelement ist der Vorgang zu wiederholen. Dabei ist die Sprache "Englisch (USA)" auszuwählen.

Als Eigenschaftenwerte werden die sogenannten "Locale Identifier" (LCID) angezeigt. Jede Sprache hat unter Windows einen solchen Schlüssel. Für Deutsch wird hier "1031" angezeigt, für Englisch "1033".

Der Testlauf zeigt, dass über die Schaltflächen die Sprache in dem 'Label'-Steuerelement umgeschaltet wird.

Texte sind in ihrem Sprachumfang erweiterbar. VisiWinNET geht aber davon aus, dass alle Texte gemeinsam erweitert werden. Über den Texteditor können über das Menü "Sprache" neue Sprachen zum Projekt hinzugefügt werden.

Über einen Doppelklick im Projektexplorer auf den Knoten "MyFirstForm" unter der Sprachumschaltung ist der Tabelleneditor wieder zu aktivieren. Das Menü "Sprache→Neue Sprache" ist auszuwählen.



Der Dialog beinhaltet die drei wichtigen Informationen für eine Projektsprache:

Sprache — Legt die Sprache und damit die LCID fest. Hier werden alle vom System unterstützten Sprachen aufgelistet. Nicht aufgelistete Sprachen (fernöstlich) müssen ggf. in der Systemsteuerung nachinstalliert werden.

Eingabegebiets-schema — Legt die Eingabemethode fest. Eine Eingabemethode enthält Informationen zur Eingabe von Zeichen über Tastatur oder IME-Hilfseditoren. Ggf. sind in der Systemsteuerung entsprechende Tastaturlayouts oder Hilfskomponenten für die gewählte Sprache nachzuinstallieren.

Editorfont — Legt die Schriftart für die Sprache im Editor fest.

Als Sprache ist hier dänisch auszuwählen. Die Auswahl ist zu bestätigen. Es ist in die Systemsteuerung des Rechners zu wechseln. (Windows-Startmenü→Einstellungen→Regions- und Sprachoptionen→Karteikarte "Sprachen"→Schaltfläche "Details"). Hier ist die Sprache "Dänisch" hinzuzufügen. Gegebenenfalls sind dazu auf Anfrage Komponenten von der Windows-CD nachzuinstallieren.

Nach erfolgreicher Installation des dänischen Eingabegebietsschemas ist im SMART die neue Sprachspalte "Dänisch" zu selektieren und über "Sprache→Sprache konfigurieren" der Dialog erneut aufzurufen. Hier ist als Eingabegebietsschema dänisch auszuwählen.

Die folgende Einstellung zeigen Sie bitte keinem Dänen.

In der dänischen Spalte des Textes 'lblWelcome' ist das Wort "Smärebräd" einzugeben.

Das "ä" wird dabei durch das dänisch-typische Schriftzeichen ersetzt. Die Einstellung eines Eingabegebietsschemas ermöglicht also Zeichen, die sonst über die Tastatur nicht zu erreichen sind.

Die Einstellung des Editorfonts in der Sprachkonfiguration wird dann nützlich, wenn fernöstliche Sprachen konfiguriert werden. Meist kann ein Unicode.Zeichensatz angegeben werden. Allerdings ist es häufig notwendig, die Schriftgröße hoch zu setzen, wenn z.B. chinesische Schriftzeichen im Editor zu klein wirken.

Um das Beispiel hier zu komplettieren, ist auf dem Formular "MyFirstForm" ein drittes 'CommandButton'-Steuerelement zu platzieren und über die 'Command'-Eigenschaft auf die neue Sprache "Dänisch" zu konfigurieren.

Wenn beispielsweise für die dänische Sprache eine besondere Darstellung bezüglich der Schriftart nötig wird, kann dies mithilfe von den Definitionstypen "Fonts" und "Fontklassen" erfolgen.

Im Projektexplorer ist der Knoten "Fontklassen" zu markieren

Die im Tabelleneditor dargestellten Fontklassen haben einen Namen und beinhalten für jede Projektsprache eine Schriftart.

In der Fonklassentabelle ist durch das Hinzufügen einer Sprache ebenfalls eine neue Spalte "Dänisch" hinzugefügt worden. Die Werte sind nicht ausgefüllt, können jedoch mit den Fontdefinitionen belegt werden.

Ist ein Steuerelement mit einer Fontklasse verknüpft, wechselt es bei einer Sprachumschaltung zu der entsprechenden Schriftart.

Das Feld "Dänisch" der Fontklasse "Labels" ist zu markieren. In der Auswahlliste des Feldes ist die Schriftart "Tahoma8BoldWestern" auszuwählen.

Auf dem Formular "MyFirstForm" ist das 'Label'-Steuerelement zu markieren. Über die ...-Schaltfläche der 'FontClass'-Eigenschaft ist der Dialog zur Auswahl einer Fontklassendefinition zu öffnen. Hier ist die Fontklasse "Labels" auszuwählen. Das Projekt ist zu starten.

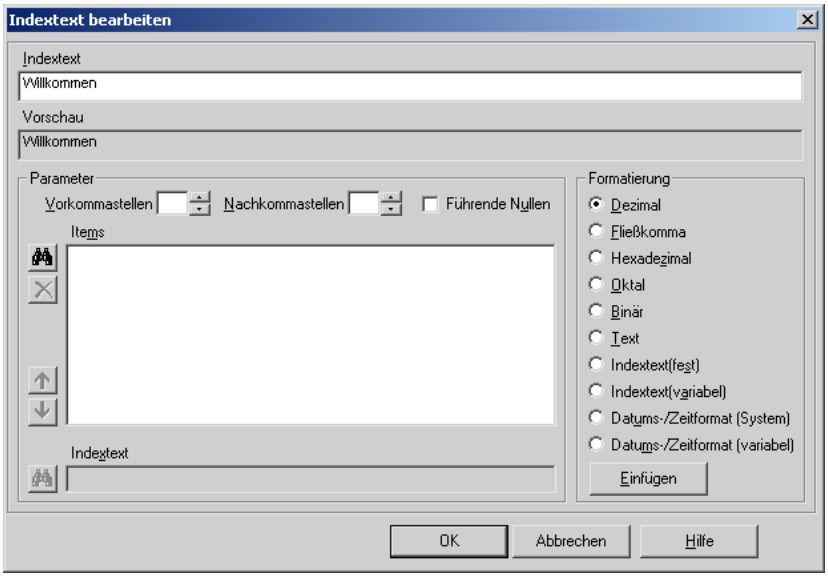
Der Testlauf zeigt, dass sich nun auch die Schriftart ändert, wenn zur dänischen Sprache gewechselt wird.

Ein weiteres Feature ist das dynamische Einbinden von Zusatzinformationen in Texte:

- Prozessvariablenwerte
- Andere Texte
- Datums-/Zeitformate

Zur Entwicklungszeit werden durch "@"-Zeichen umfasste Formatzeichenketten in den Text eingefügt. Zur Laufzeit werden diese Bereiche mit den entsprechenden Werten ersetzt.

Der Text "lblWelcome" ist im Tabelleneditor sichtbar zu schalten. Über die ...-Schaltfläche in der Spalte "Deutsch" ist der Dialog zum Bearbeiten des Textes zu öffnen.



Dieser Dialog enthält einen Formatzeichenkettengenerator, dessen Funktionen das Einfügen der o.g. Funktionen erlaubt.

Über die -Schaltfläche links neben der "Items"-Liste ist der Variablenauswahldialog zu öffnen. Hier ist ein Item auszuwählen.

In der Formatierungsliste ist der Eintrag "Dezimal" zu wählen. Über die "Einfügen"-Schaltfläche ist die Formatzeichenkette in den Text einzufügen.

Nachdem die Einstellungen über die OK-Schaltfläche bestätigt wurden, steht im Textfeld folgender Text:

"Willkommen @1d@"

Die in "@"-Zeichen eingefasste Formatierungszeichenkette gibt an, dass der erste in der Prozessvariablenliste angegebene Prozesswert als Dezimalwert an dieser Stelle ausgegeben werden soll.

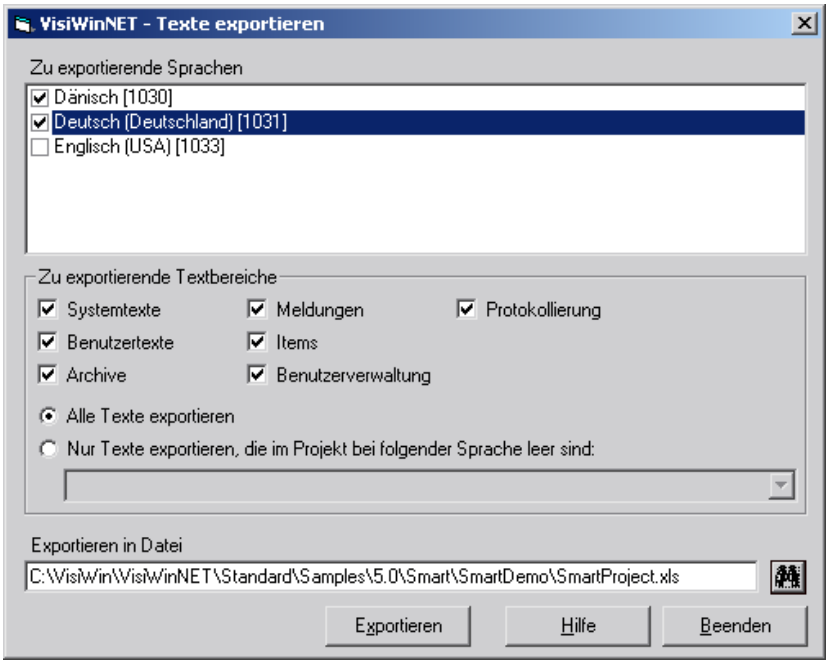
Der Testlauf zeigt, dass tatsächlich der entsprechende Wert der Variablen im Text eingeblendet wird.

Neben dem 'Label'-Steuerelement unterstützen auch andere Steuerelemente direkt oder indirekt die Anzeige von sprachumschaltbaren Texten. Generell gilt: Eine 'LocalizedText'-Eigenschaft kennzeichnet die Anzeige eines sprachumschaltbaren Textes.

Andere Steuerelemente haben keine solche Eigenschaft, beziehen aber Texte oder Formatierungsanweisungen aus der Sprachumschaltung. Das 'AlarmLine'-Steuerelement beispielsweise beinhaltet die Eigenschaft 'DateTimeFormat', die die Angabe eines Textes aus der Sprachumschaltung erlaubt. Die Angabe "@LongDate" als Eigenschaftenwert besagt, dass der gleichnamige Text aus der Textgruppe "Komponenten.Time.DateFormats" für jede Projektsprache eine Formatierungszeichenkette enthält, die zur Laufzeit dazu verwendet wird, Datums-/Zeitwerte zu formatieren. Auch dies ist eine Funktion der Sprachumschaltung.

Früher oder später wird der Projekteur in die Verlegenheit kommen, Texte für ein Übersetzungsbüro exportieren zu müssen. Die entsprechenden Funktionen sind im Menü "Extras" des Texteditors enthalten.

Der Menüeintrag "Extras→Übersetzung→Export" ist zu wählen.



Der Dialog zum Exportieren der Texte wird geöffnet. Als Ausgabeformat stehen xls- (Excel) oder mdb- (Access 2000) zur Verfügung. Die so erzeugte Datei kann dem Übersetzungsbüro zugesendet werden.

Nach der Übersetzung können die Texte über den Menüeintrag "Extras→Übersetzung→Import" wieder reimportiert werden.

2.1.6 Meldesystem

Das Meldesystem dient zur zentralen Erfassung und Speicherung von Fehlerzuständen oder spezifischen Meldungen. Die entsprechenden Definitionen, welche Variable eine Meldung auslöst, wie die Meldung dargestellt wird und welche Zusatzinformationen zur Meldung gehören, werden im Meldeeditor projektiert. Die Variablen, die die Meldungen auslösen, werden von VisiWinNET beim Variablenkern zur Überwachung angemeldet. Bei einer Variablenwertänderung wird die entsprechende Meldung zur Applikation übertragen, wo sie in speziellen Steuerelementen angezeigt wird.

VisiWinNET beinhaltet zwei Steuerelemente, die die direkte Anzeige anstehender Meldungen erlauben:

AlarmList Alle anstehenden Meldungen werden in Form einer Liste angezeigt. Das Steuerelement unterstützt dabei zusätzlich die Quittierung von Meldungen.

AlarmLine Die platzsparende Variante, die auf allen Formularen im Kopf- oder Fußbereich Einsicht in anstehende Meldungen erlaubt.

Zusätzlich zur Übertragung der Meldedaten in die Oberfläche erlaubt VisiWinNET auch die Speicherung von Meldungen in Dateien. Dadurch ist ein Rückblick auf den Produktionsverlauf einer Anlage möglich. Das Steuerelement 'HistoricalAlarmList' dient zur Anzeige dieser aufgezeichnete Daten in Form einer Liste.

Eine Meldung wird durch die Änderung eines Bits aus dem Variablenkern gesteuert. Wählbar ist, ob das Bit auf "1" oder "0" gesetzt wird, damit die Meldung ausgelöst wird. Über eine Erweiterung ist es aber auch möglich, analoge Größen beispielsweise auf

Grenzwertüberschreitungen zu überprüfen. Meldungen können vom Benutzer quittiert werden. Eine Quittierrückmeldung an die Steuerung ist ebenfalls realisierbar.

Das Erscheinungsbild von Meldungen in den Steuerelementen 'AlarmLine' und 'AlarmList' wird über Meldeklassen parametriert. Meldeklassen beinhalten verschiedene Parameter (Farben, Statustexte und Symbole), die die Anzeige einer Meldung beschreiben. Jede Meldung referenziert eine Meldeklasse. So stehen jeder Meldung die Anzeigeparameter einer Meldeklasse zur Verfügung.

Meldegruppen gliedern Meldungen. Ähnlich wie in der Sprachumschaltung ist es sinnvoll, die einzelnen funktionalen Elemente einer Anlage zu gruppieren. Dies fördert die Übersichtlichkeit und macht das Projekt modularer. Meldegruppen beinhalten aber auch zusätzlich Parameter, die die gruppenweise Deaktivierung oder Quittierung von Meldungen zulassen.

Das folgende Beispiel zeigt:

- Das generelle Projektieren, Auslösen und Anzeigen von Meldungen, die als Bits im Variablenkern vorliegen
- Das Quittieren von Meldungen
- Verschiedene Anzeigefilter
- Die Überwachung von analogen Werten

Als erstes sind die Quellen der Meldungen zu definieren. Da Meldungen variablengebunden sind, beginnt also alles in der Prozessanbindung. Da allerdings keine realen Steuerungsdaten zur Verfügung stehen, müssen die auslösenden Variablen über interne Variablen simuliert werden.

Im Projektexplorer ist der Knoten "Items" zu expandieren. Über den Knoten "Intern" ist der Tabelleneditor zu öffnen. Hier sind folgende Items zu definieren:

Name	Datentyp
MessageSource	VT_I2
MessageState	VT_I2
GroupFunction	VT_I2

Als funktionale Datenanbindung reichen die hier definierten Variablen:

- MessageSource beinhaltet die Bits, die die Meldungen auslösen
- MessageStates beinhaltet die von der Kernfunktionalität zurückgegebenen Meldezustände
- GroupFunction dient zur Steuerung von Meldungsfunktionalitäten über die übergeordnete Gruppe

Nun folgt die Festlegung der Definitionen im Meldesystem:

Der Knoten "Meldungen" im Projektexplorer ist zu expandieren. Das Kontextmenü des darunter liegenden Knotens "Meldegruppen" ist über einen Klick mit der rechten Maustaste aufzurufen. Über den Eintrag "Neu" ist eine Meldegruppe zu erzeugen. In der Meldegruppe sind folgende Parameter einzustellen:

Parameter	Wert
Name	Grp1
Quittierungsvariable	GroupFunction
Bitnummer Quittierungsvariable	0

Eine zweite Meldegruppe ist unter dem Knoten "Meldegruppen" zu erzeugen

Vorsicht: Meldegruppen sind schachtelbar. Daher zuerst zum Knoten "Meldegruppen" wechseln

Die Parameter der zweiten Meldegruppe sind auf folgende Werte einzustellen:

Parameter	Wert
Name	Grp2
Quittierungsvariable	GroupFunction
Bitnummer Quittierungsvariable	1

Der Tabelleneditor ist durch einen Doppelkilck auf den Knoten Grp1 zu öffnen. Die leere Tabelle ist durch einen Klick auf die freie Fläche zu markieren. Anschließend sind über die Taste "F8" zwei neue Meldungen anzulegen. Nachdem die Eingabemarke auf einen Datensatz im Tabelleneditor gesetzt wurde, ist über die Taste "F9" der Dialog zum Einstellen der Parameter der Meldedefinitionen zu öffnen. Folgende Einstellungen sind hier vorzunehmen:

Für die 1. Meldung

Parameter	Wert
Name	Msg1
Ereignisvariable	MessageSource
Bitnummer	0
Meldetext	MessageSource Bit 0
Priorität	1
Statusvariable	MessageState
Bitnummer Statusvariable	0

Für die 2. Meldung

Parameter	Wert
Name	Msg2
Ereignisvariable	MessageSource
Bitnummer	1
Meldetext	MessageSource Bit 1
Priorität	2
Statusvariable	MessageState
Bitnummer Statusvariable	4

Für die Auswahl von Variablen steht auf dem Dialog jeweils die -Schaltfläche rechts neben dem entsprechenden Parameterfeld zur Verfügung.

Zwei weitere Meldungen sind unter "Grp2" anzulegen. Auf die gleiche Weise wie oben sind die Parameter dieser Meldungen anzupassen:

Für die 1. Meldung

Parameter	Wert
Name	Msg3
Ereignisvariable	MessageSource
Bitnummer	2
Meldetext	MessageSource Bit 2
Priorität	3
Statusvariable	MessageState
Bitnummer Statusvariable	8

Für die 2. Meldung

Parameter	Wert
Name	Msg4
Ereignisvariable	MessageSource
Bitnummer	3
Meldetext	MessageSource Bit 3
Priorität	4
Statusvariable	MessageState
Bitnummer Statusvariable	12

Die oben angegebenen Werte werden im Folgenden die volle Funktion des Meldesystems zeigen. Der oben projektierten Parameter "Statusvariable" ist optional. Die Festlegung eines Meldetextes ist quasi Pflicht. Die Angabe der Ereignisvariable schließlich muss sein.

Alle neu angelegten Meldungen zeigen mit dem Parameter "Meldeklasse" auf den Wert "Alarm". Dahinter steckt eine Definition, die sich unter dem Knoten "Meldeklassen" des Projektexplorers verbirgt.

Der Knoten "Meldeklassen" im Projektexplorer ist zu markieren. Im Tabelleneditor ist die Eingabemarke auf die "Alarm"-Meldeklase zu setzen und über die Taste "F9" der Dialog für die Meldeklasse zu öffnen. Hier ist der Parameter "Quittierungsart" auf "quittierpflichtig" zu setzen.

VisiWinNET unterstützt verschiedene Quittierungsarten. Eine Quittierungsart legt fest, wie oft eine Meldung vom Benutzer oder von der Steuerung quittiert werden muss, bevor sie aus den Steuerelementen der Applikation verschwindet. Durch diese Einstellung kann eine Meldung zur Laufzeit verschiedene Stati einnehmen. Beispielsweise unterscheidet das System zwischen dem Zustand "gekommen" (Ereignisbit wurde gesetzt) und "gekommen quittiert" (Nach dem Setzen wurde vom Benutzer das Auftreten quittiert, das Ereignisbit ist aber noch nicht zurückgesetzt worden). Jeder Zustand kann durch ein Symbol, einen Statustext und Farben kenntlich gemacht werden. Auch diese Festlegungen erfolgen in den Meldeklassen.

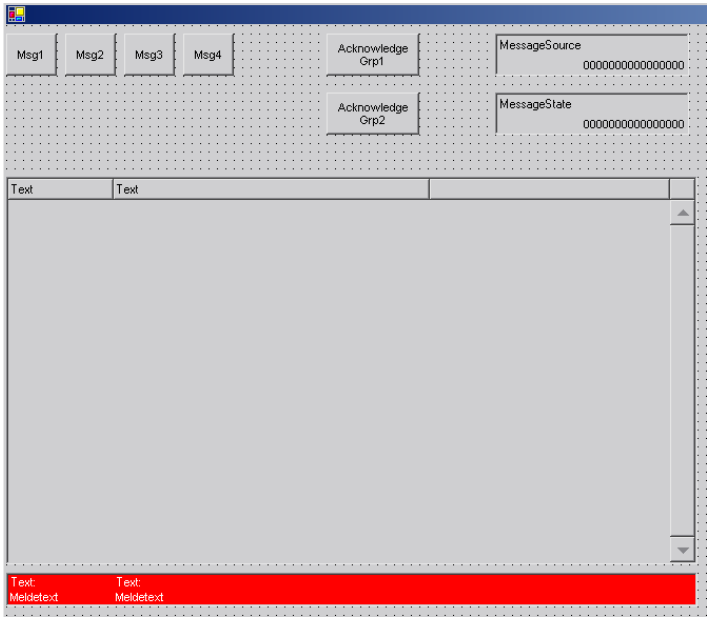
Der Status einer Meldung kann auch bitcodiert in einer Variablen hinterlegt werden. Dazu sind in den Meldungen die Statusvariablen mit Bitnummern festgelegt worden. Jeweils drei Bits repräsentieren den Status einer Meldung.

Damit auch alle definierten Information in der Anzeige erscheinen, müssen nun die Oberflächenelemente projektiert werden. Der Übersicht halber erfolgt dies in einem eigenen Formular.

Über den Knoten "Design" im Projektexplorer ist ein neues Formular "MyAlarmForm" anzulegen. Mithilfe von "CommandButton"-Steuerelementen sind die entsprechenden Seitenwechsel zu dem Formular zu implementieren.

Auf diesem Formular sind folgende Steuerelemente (von oben nach unten, siehe Grafik) zu platzieren:

- 2 x VarOut
- 4 x Switch
- 2 x Key
- 1 x AlarmList
- 1 x AlarmLine



Die vier 'Switch'-Steuerelemente sind über die 'VWItem'-Eigenschaft mit der Variablen "MessageSource" zu verbinden. Die 'BitNumber'-Eigenschaft im VWItem-Objekt ist dabei für die einzelnen Steuerelemente auf die Werte 0, 1, 2 und 3 zu setzen. Dies sind die Bits, die die Meldungen auslösen. Über die Text-Eigenschaft kann die Funktion der Schaltfläche angegeben werden.

Die beiden 'Key'-Steuerelemente sind über die 'VWItem'-Eigenschaft an die Variable "GroupFunction" anzubinden. Die 'BitNumber'-Eigenschaft im 'VWItem'-Objekt ist dabei für das zweite Steuerelement auf den Wert 1 zu setzen. Die beiden angeschlossenen Bits sind die Quittierbits der Meldegruppen. Auch hier kann über die 'Text'-Eigenschaft die Funktion der Schaltflächen abgebildet werden.

Die zwei 'VarOut'-Steuerelemente sind über die VWItem-Eigenschaft mit den beiden internen Variablen "MessageSource" und "MessageState" zu verknüpfen. Zusätzlich ist die 'Format'-Eigenschaft auf 'binary' einzustellen. Über die Eigenschaft Label→Text→Text ist der angebundene Variablename als Informationstext einzugeben.

Die Eigenschaften der Steuerelemente 'AlarmList' und 'AlarmLine' müssen für den ersten Testlauf nicht angepasst werden.

Die Applikation ist zu starten. Die vier unteren Bits der Variablen "MessageSource" sind über die entsprechenden 'Switch'-Steuerelemente zu setzen.

Folgende Änderungen werden sichtbar:

- In der Meldeliste werden alle vier Meldungen angezeigt
- In der Meldezeile wird die Meldung "Msg4" angezeigt. Dies ist die aktuell im System anstehende Meldung mit der höchsten Priorität
- Die Variable "MessageState" zeigt jeweils für jede Meldung den Status "100" (binär) an. Nötigenfalls könnte dieser Status auch an die Steuerung zurück übertragen werden.

Die vier unteren Bits der Variablen "MessageSource" sind über einen erneuten Klick auf die 'Switch'-Steuerelemente wieder auf "0" zu setzen.

- Die Meldezeile zeigt keine Meldunge mehr
- In der Meldeliste bleiben die Meldungen erhalten. Es ändert sich jedoch der Status und damit die Farbe. Durch die gewählte Quittierungsart ist eine Quittierung nötig, um die Meldungen endgültig zurückzusetzen.
- Die Statusbits von Meldungen und Meldegruppen zeigen jetzt den Status "010"

Die Meldungen sind nacheinander durch einen Doppelklick auf die entsprechenden Einträge in der Meldeliste zu quittieren.

- Die Meldungen verschwinden aus der Meldeliste.
- Auch die Statusbits werden in "MessageState" zurückgesetzt

Die Meldungen sind erneut über die 'Switch'-Steuerelemente zu generieren.

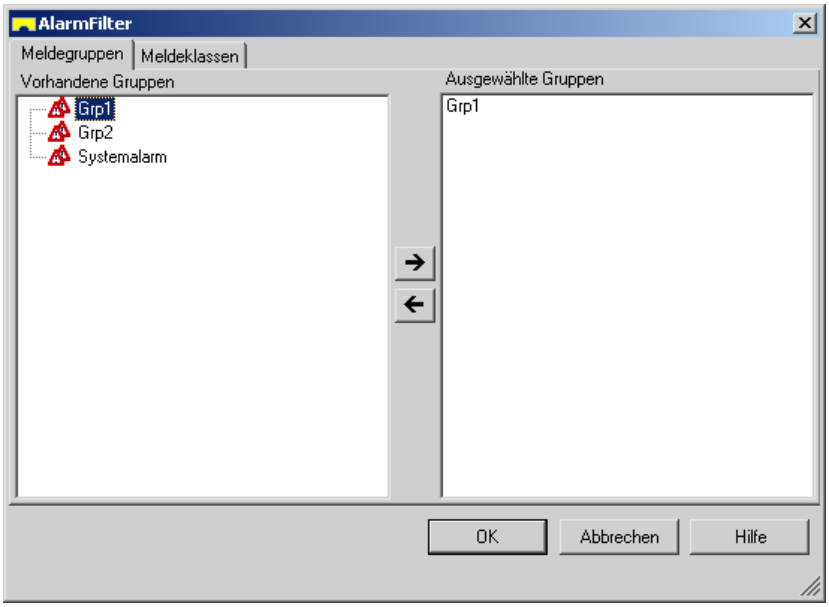
Die beiden 'Key'-Steuerelemente sind zu drücken.

Über das Quittierbit einer Gruppe werden die enthaltenen Meldungen gemeinsam quittiert.

Hier kann das Projekt gestoppt werden.

VisiWinNET enthält die Möglichkeit, die Anzeige von Meldungen in den Steuerelementen zu filtern. Dadurch ist es beispielsweise möglich, eine Meldeliste nur mit Meldungen über Anlagenstörungen zu projektieren und eine weitere mit Produktionshinweisen. Im Client-Server-System wird dadurch auch möglich, nur die für den Arbeitsplatz wichtigen Meldungen anzuzeigen und trotzdem eine zentrale Datenhaltung zu behalten.

Zurück in der Designansicht ist das 'AlarmList'-Steuerelement zu markieren. Über die -Schaltfläche der 'AlarmFilter'-Eigenschaft ist der Dialog zur Filterfestlegung zu öffnen.



Der Filter erlaubt die Auswahl von Meldegruppen und Meldeklassen. Nur die Meldungen, die in den hier ausgewählte Gruppen liegen, werden im Steuerelement angezeigt.

Über einen Doppelklick auf den Eintrag "Grp1" ist die gleichnamige Gruppe auszuwählen. Nach Bestätigen der Auwahl über die OK-Schaltfläche ist das Projekt erneut zu starten. Noch einmal sind alle vier Meldungen über "MessageSource"zu generieren.

Nur die in "Grp1" liegenden Meldungen werden in der Meldeliste angezeigt. Die Statusbits zeigen aber, dass auch die in "Grp2" liegenden Meldungen generiert wurden. Der Filter beschränkt sich also wirklich nur auf die Anzeige im Steuerelement.

Nachdem also hoffentlich Ursache und Wirkung von Meldungen hinreichend geklärt sind, folgt nun die Beschreibung der Analogwertüberwachung. Ein typisches Beispiel ist eine Temperatursensor, dessen Wert in unterschiedliche Bereiche (Normal, Kritisch, Fehler) eingeteilt wird.

Die folgende Beschreibung kann nicht für "Compact"- oder "Embedded"-Projekte ausgeführt werden.

Im Projektexplorer ist der oberste Knoten mit dem Projektnamen zu markieren. Über das Kontextmenü ist der Eintrag VisiWinNET-Eigenschaften auszuwählen. Auf dem folgenden Dialog ist die Karteikarte "Serverkomponenten" sichtbar u schalten. In der Liste der Serverkomponenten ist der Eintrag "Supervision/Prozessdatenüberwachung" über das Kontrollkästchen zu aktivieren.

Im Projektexplorer wird ein neuer Knoten mit der Bezeichnung "Prozessdatenüberwachung" sichtbar.

Die Vorgehensweise zeigt, dass VisiWinNET aus einzelnen Komponenten besteht, die individuell in einem Projekt aktiviert oder deaktiviert werden können. Eine deaktivierte Komponente wird zur Laufzeit nicht geladen.

Zwei neue interne Items mit dem Datentyp "VT_I2" sind im Itemeditor anzulegen. Die Namen der beiden Items sollen "AnalogValue" und "SupervisionOut" sein.

Im Editor des Meldesystems ist eine neue Meldegruppe mit dem Namen "AnalogValueAlarm" anzulegen. In dieser Gruppe sind drei Meldungen zu definieren, die folgende Parameterwerte haben:

Parameter	Meldung 1	Meldung 2	Meldung 3
Name	Msg1	Msg2	Msg3
Text	AnalogValue Warnung 1	AnalogValue Warnung 2	AnalogValue Alarm
Ereignis-variable	SupervisionOut	SupervisionOut	SupervisionOut
Bitnummer Ereignis-variable	3	4	5

Über das Kontextmenü des Knoten "Prozessdatenüberwachung" ist eine neue Überwachungsgruppe anzulegen. Auf dem Dialog, der die Parameter der Überwachungsgruppe beinhaltet, sind folgende Einstellungen vorzunehmen:

Parameter	Wert
Name	Analog
Typ	Bereichsüberwachung

Über einen Doppelklick auf den neuen Knoten im Projektexplorer ist der Tabelleneditor der Prozessdatenüberwachung zu öffnen. Die leere Tabelle ist durch einen Klick zu markieren. Über die Taste "F8" ist eine neue Überwachung hinzuzufügen. Die Parameter dieser Überwachung sind mit folgenden Werten zu füllen:

Parameter	Wert
Name	AnalogValueSupervision
Überwachungs-variable	AnalogValue
Ausgabevariable	SupervisionOut
Bitnummer Ausgabevariable	0
Grenze HHH	100
Grenze HH	90
Grenze H	70

Das Formular "MyFirstAlarm" ist zu öffnen. Hier ist ein VarIn-Steuerelement hinzuzufügen und über die 'VWItem'-Eigenschaft mit der Variablen "AnalogValue" zu verbinden.

Zuletzt ist der Gruppenfilter der Alarmliste, der in den vorhergehenden Schritten eingestellt wurde, wieder zu entfernen (alle Gruppen wieder rausnehmen).

Das Projekt ist zu starten. Der Wert der "AnalogValue"-Variablen ist über das entsprechende 'VarIn'-Steuerelement auf die in der Überwachung projektierten Grenzwerte "70/90/100" zu setzen.

Die entsprechenden Meldungen aus dem Meldesystem werden ausgelöst. Konzeptionell sind überwachte Analogwerte also keine eigenständigen Meldungen. Die Prozessdatenüberwachung prüft nur Prozesswerte auf Grenzwertüberschreitung und setzt entsprechend Bits in anderen Prozesswerten. Diese dienen dann dazu, bitgesteuerte Meldungen auszulösen.

Vorsicht Falle: Sollten die Meldungen in der Meldeliste nicht angezeigt werden, liegt das an dem im vorhergehenden Schritt festgelegten Gruppenfilter.

In der Meldezeile sollte aber auf jeden Fall die Meldung sichtbar sein.

2.1.7 Archivsystem

Das VisiWinNET Archivsystem erlaubt die Aufzeichnung von Prozesswerten. Als Ansichtskomponente steht das 'TrendChart'-Steuerelement zur Verfügung. Die nötigen Definitionen, welche Prozesswerte wie aufgezeichnet werden sollen, werden im Archiveditor vorgenommen.

Im Projektexplorer ist der Knoten "Archive" zu markieren. Über das Kontextmenü ist eine neue Archivdefinition anzulegen. Über den Eintrag "Bearbeiten" im Kontextmenü des neuen Archives ist der Dialog zur Festlegung der Archivparameter zu öffnen. Hier sind folgende Parameter anzupassen:

Parameter	Wert
Name	Archiv1
Typ (Karteikarte "Format")	Folgearchiv
Abtastzeit (Karteikarte "Aufzeichnung")	10

Ein Archiv ist ein Container für aufzuzeichnende Werte Dieser Container ...

- ... fasst mehrere aufgezeichnete Prozesswerte in Dateien zusammen
- ... legt das Abtastverhalten fest
- ... administriert die erzeugten Dateien

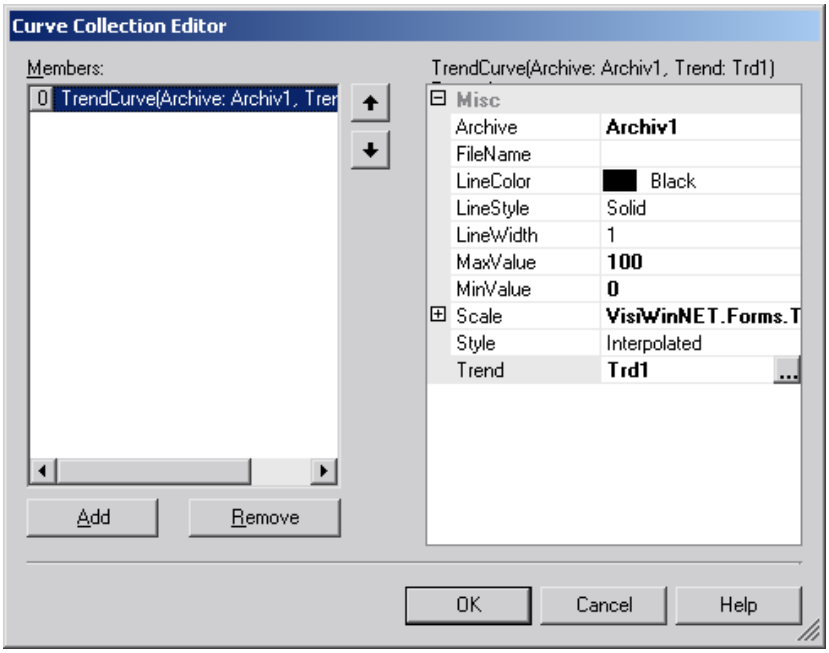
Die Festlegung der aufzuzeichnenden Prozesswerte wird im folgenden Schritt beschrieben.

Über einen Doppelklick auf das neue Archiv im Projektexplorer ist der Tabelleneditor zu öffnen. Die leere Tabelle ist durch einen Klick zu markieren. Über "F8" ist eine neue Trenddefinition anzulegen. Über die -Schaltfläche in der Spalte "Trendvariable" ist der Variablenauswahldialog zu öffnen. Ist in den vorhergehenden Schritten ein OPC-Server verwendet worden, ist die Variable "Ch1.DistillationDevice.Unit1.WaterTank.Temperature" auszuwählen.

Wurde in den vorhergehenden Schritten ein VisiWin-Treiber verwendet, kann hier die Variable "Ch1.DummyItem<.Saege1>" verwendet werden.

Damit ist auch die Festlegung des Prozesswertes erledigt. Es fehlt noch die Ansicht der aufgezeichneten Werte in der Applikation.

Ein neues Formular mit dem Namen "MyFirstTrend" ist zu erzeugen und über entsprechend parametrierte 'CommandButton'-Steuerelemente aus den anderen Formularen aufrufbar zu machen. Auf diesem Formular ist ein Steuerelement vom Typ 'TrendChart' zu platzieren. Über die ...-Schaltfläche der 'Curves'-Eigenschaft ist der Dialog zur Parametrierung der anzuzeigenden Kurven zu öffnen.



Die Liste auf der linken Seite enthält die zu zeichnenden Kurven. Eine Kurve ist schon im Dialog eingefügt, allerdings hat sie noch keine Anbindung an eine Trenddefinition.

Der Eintrag links ist zu markieren. Über die ...-Schaltfläche der 'Archive'-Eigenschaft rechts ist der Dialog zur Auswahl von Archiv- und Trenddefinitionen zu öffnen. In der "Archiv"-Liste ist "Archiv1" auszuwählen und in der Trendliste ist "Trd1" zu markieren. Die Auswahl ist über die "OK"-Schaltfläche zu bestätigen. Anschließend ist die Eigenschaft 'MaxValue' auf 150 einzustellen.

Die Dialogauswahl wird in die Eigenschaften 'Archive' und 'Trend' übernommen. 'MaxValue' parametriert die obere sichtbare Wertegrenze für diese Kurve.

Der Dialog ist über "OK" zu schließen. Die Eigenschaft 'TimeScale' ist zu expandieren. In der darunterliegenden 'Resolutiuon'-Eigenschaft ist der Wert "00:02:00" anzugeben. Das Projekt ist zu starten.

Die 'Resolution'-Eigenschaft legt den sichtbaren zeitlichen Bereich im Steuerelement fest. Hierüber kann also die x-Achse parametriert werden.

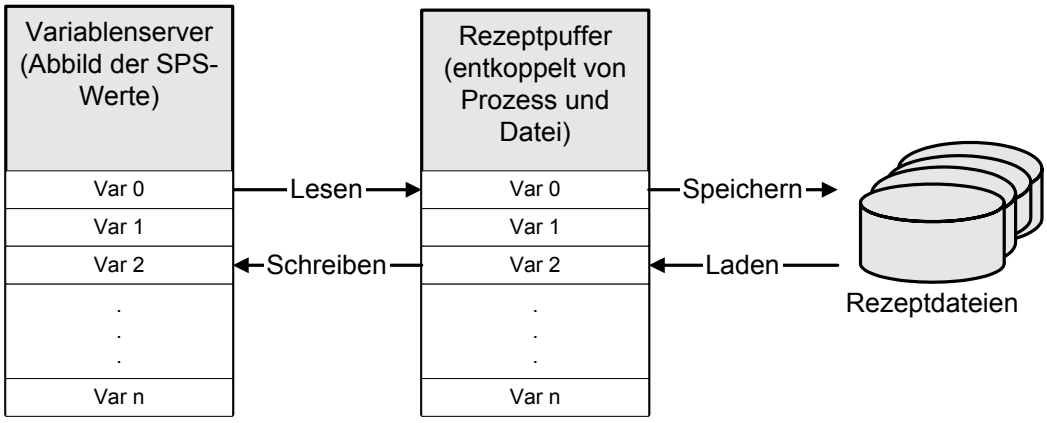
Die laufende Applikation zeigt den in der Trenddefinition angegebenen Prozesswert als Kurve.

2.1.8 Rezepte

Visualisierungsrezepte beschreiben Sätze von Prozesswerten. Diese legen üblicherweise gemeinsam die Einstellungen einer Maschine zur Fertigung eines Produktes aus einer Produktpalette fest. Über das Schreiben von Rezepten zur SPS kann damit zwischen den einzelnen zu fertigenden Produkten umgeschaltet werden.

Zur Entwicklungszeit wird über Rezeptdefinitionen festgelegt, welche Variablen die Funktion gemeinsam beschreiben, welche Variablen also gemeinsam gespeichert und transferiert werden.

Zur Laufzeit werden die Prozesswertsätze in sogenannten Rezeptdateien abgelegt. Zwischen dem Prozessabbild und den Rezeptdateien liegt der Rezeptpuffer. Über diesen Puffer können Rezeptwerte in der Applikation geändert werden, ohne dabei SPS-Daten oder Rezeptwertdaten zu beeinflussen. Der Rezeptpuffer dient zur Parametrierung bzw. zur Optimierung von Rezepten. Aus der Applikation werden beispielsweise Rezeptdaten aus einer Datei geladen, im Rezptpuffer an das Produktionsziel angepasst und anschließend zur SPS transferiert. der umgekehrte Weg ermöglicht die Kontrolle eingestellter Werte, bevor eine neue Rezeptdatei erzeugt wird.



Die vier Transferfunktionen "Laden", "Speichern", "Lesen" und "Schreiben" steuern die einzelnen Übertragungsschritte.

Typische Eingabesteuerelemente, wie z.B. das 'VarIn', unterstützen sowohl die Ansicht auf die Prozesswerte im Variablenkern als auch im Rezeptpuffer. Sollen also diese Werte in der Applikation zum Bearbeiten verfügbar gemacht werden, ist die Applikation nicht um neue Steuerelemente zu erweitern. Vielmehr ist die an eine Rezeptdefinition zentral gekoppelte Umschaltung der Ansicht (für alle in der Definition verwendeten Variablen) zu implementieren.

Weitere Funktionen, wie die FDA-konforme Erfassung von Rezeptwertänderungen oder das Speichern von Produktionskommentaren sind einfach parametrierbar.

Die typischen Aufgaben für den Entwickler im Rezeptsystem sind:

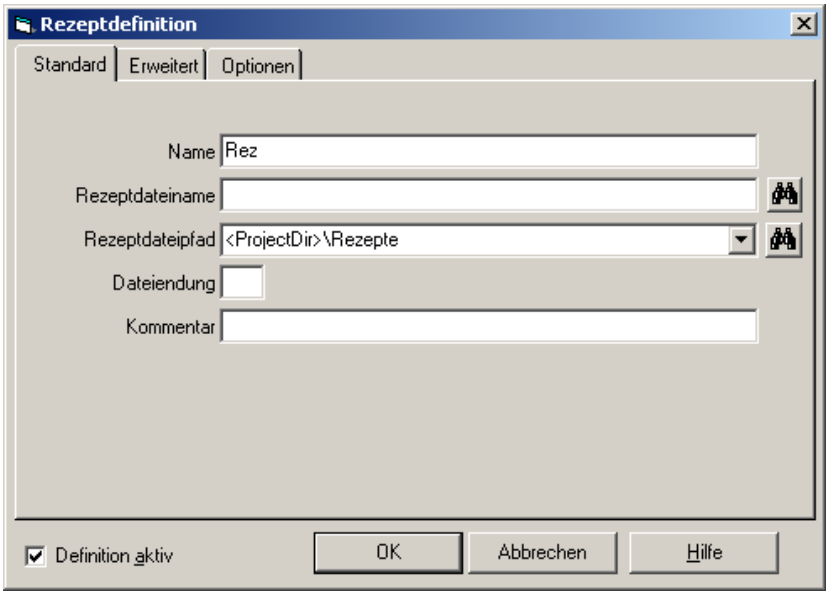
Festlegung der Rezeptdefinitionen	Hier wird festgelegt, welche Variablen das Rezept beschreiben. Um allen Irrtümern vorzubeugen: Es geht nicht um die Variablenwerte. Diese werden erst zur Laufzeit festgelegt.
Designanpassung: Editieren der Rezeptwerte	Grundsätzlich können die bestehenden Steuerelemente verwendet werden, die auf die Variablen im Prozess gebunden sind.
Designanpassung: Administration der Rezeptdateien	Typische Aufgaben, wie das Laden oder Speichern von Rezeptwerten aus Dateien, erfordern meist eine Auflistung der bestehenden Rezeptdateien.

Das folgende Beispiel veranschaulicht die o.g. Schritte anhand eines Minimalrezeptes aus zwei Variablen.

Im Projektexplorer ist der Knoten "Items" zu expandieren. Über den Knoten "Intern" ist der Tabelleneditor zu öffnen. Hier sind folgende Items zu definieren:

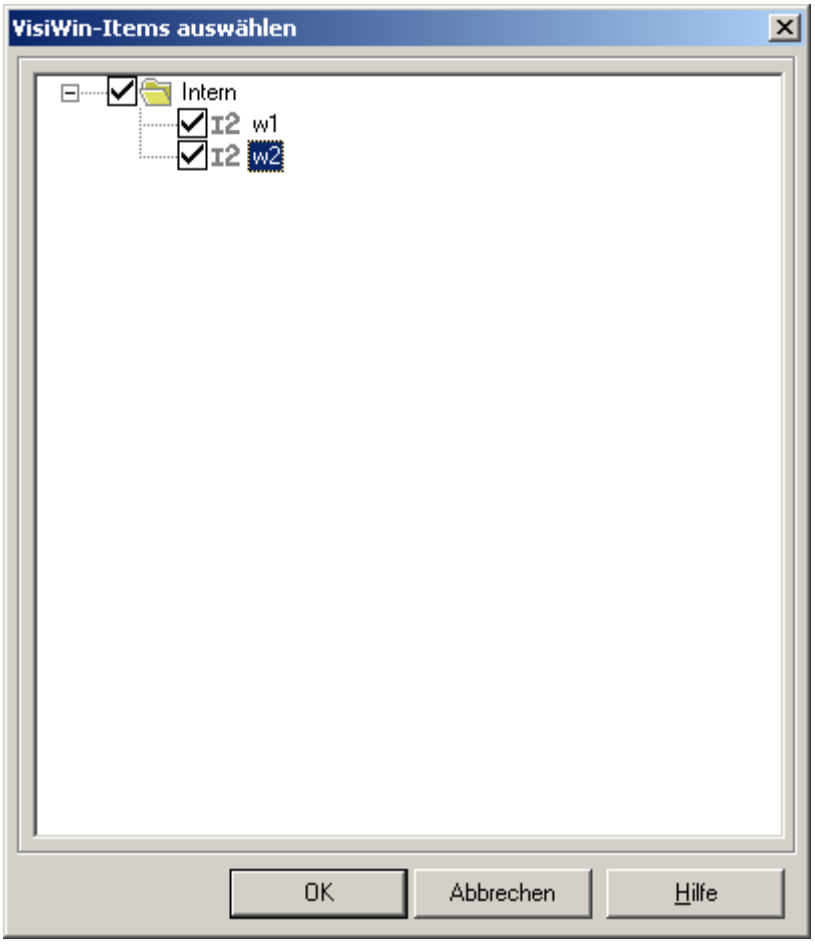
Name	Datentyp
w1	VT_I2
w2	VT_I2

Im Projektexplorer ist der Knoten "Rezepte" zu markieren. Über das Kontextmenü des Knotens ist eine neue Rezeptdefinition zu erzeugen. Der Name der Definition ist über den Menüpunkt "Bearbeiten" im Kontextmenü des neuen Knotens auf "Rez" einzustellen.



Nach der Bestätigung des Dialoges über die OK-Schaltfläche ist der Tabelleneditor der Rezeptdefinition über einen Doppelklick auf den Knoten zu öffnen.

Über den Menüpunkt Rezeptelemente→Variablen hinzufügen/löschen ist der Variablenauswahldialog zu öffnen.

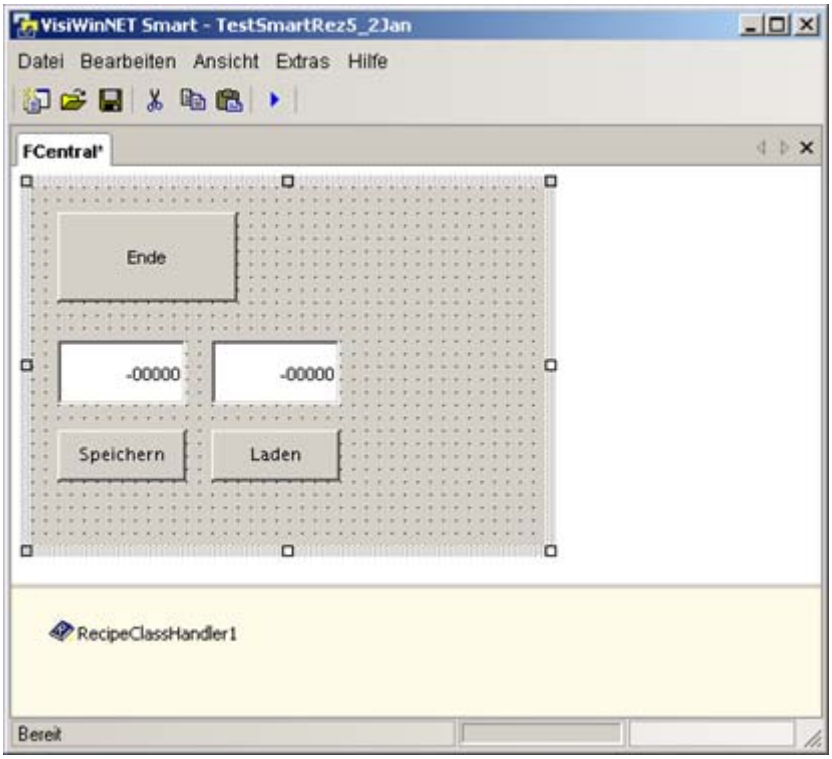


Hier werden die Variablen festgelegt, die zur Rezeptdefinition gehören.

Die beiden interne Variablen sind zu markieren. Der Dialog ist über die OK-Taste zu verlassen.

Die markierten Items werden nun als Elemente des Rezeptes im Tabelleneditor aufgeführt.

Auf einem Formular sind zwei 'VarIn', zwei 'CommandButton' und eine 'RecipeClassHandler'-Komponente zu platzieren (Siehe Grafik):



Nachdem die RecipeClassHandler-Komponente auf dem Formular platziert wurde, rutscht sie automatisch in das sogenannte "Komponentenfach" unterhalb der Designansicht des Formulars.

Folgende Eigenschaftenwerte sind hier zu ändern:

Für VarIn1

Eigenschaft	Wert
VWItem	w1

Für VarIn2

Eigenschaft	Wert
VWItem	w2

Für CommandButton1

Eigenschaft	Wert
Text	Speichern
Events	Das Click-Ereignis ist mit dem Aufruf der 'GetRecipe'-Funktion der 'RecipeClassHandler'-Komponente zu verknüpfen. Über die ...-Schaltfläche in der 'Events'-Eigenschaft öffnet sich der Dialog 'Ereignisverarbeitung'. Hier ist das Click-Ereignis auszuwählen und die "Hinzufügen"-Schaltfläche zu drücken. Auf dem nun sichtbaren Funktionsauswahldialog ist die o.g. Funktion auszuwählen. Nach dem Schließen des Funktionsauswahldialoges ist sicherzustellen, dass die Funktion in der Funktionenliste erscheint und ihr Parameter 'Provider' auf die 'RecipeClassHandler'-Komponente des Formulars verweist.

Für CommandButton2

Eigenschaft	Wert
Text	Laden
Events	Das Click-Ereignis ist mit dem Aufruf der 'ShowLoadRecipeDialog'-Funktion der 'RecipeClassHandler'-Komponente zu verknüpfen (s.o.).

Für RecipeClassHandler1

Eigenschaft	Wert
RecipeClass	Rez (Der Name der Rezeptdefinition aus der Projektdatenbank)
Events	Folgende Ereignisse werden hier verwendet: • Das 'GetDoneSucceeded'-Ereignis ist mit dem Aufruf der 'ShowSaveRecipeDialog'-Funktion der 'RecipeClassHandler'-Komponente zu verknüpfen. • Das 'LoadDoneSucceeded'-Ereignis ist mit dem Aufruf der 'SetRecipe'-Funktion der 'RecipeClassHandler'-Komponente zu verknüpfen.

Die so parametrierten Ereignisse bewirken beim Speichern folgende Abläufe:

- Das Klick-Ereignis triggert die Funktion 'GetRecipe' an. Dadurch werden die Werte der als Rezeptelemente festgelegten Variablen in den Rezeptpuffer geladen.
- Den Abschluss des Einlesens meldet der RecipeClassHandler im 'GetDoneSucceeded'-Ereignis. Dieses wiederum ist mit der 'ShowSaveRecipeDialog'-Funktion. Zur Laufzeit wird hierüber das Speichern der Rezeptdatei geregelt, wobei vor dem Speichern ein Dialog zur Eingabe des Namens und der Beschreibung angezeigt wird.

Der umgekehrte Weg wird über die Laden-Schaltfläche ausgelöst. Zuerst wird über einen Auswahldialog eine Rezeptdatei in den Rezeptpuffer geladen und anschließend von dort aus in den Prozess transferiert.

Dieses Beispiel verwendet den Rezeptpuffer praktisch nicht. In vielen Anwendungen mag es ausreichen, wenn Rezepte direkt im Prozess erstellt und optimiert werden.

Die Verwendung des Rezeptpuffers wird aber dann sinnvoll, wenn Produktionseinstellungen über die Applikation vorzunehmen sind, die während des Einstellens die Werte im Prozess nicht verändern dürfen. Erst nach Abschluss der Einstellungen können die Daten dann gemeinsam in den Prozess übertragen werden.

Im Projekt "SmartDemo" wird beispielhaft die Verwendung des Rezeptpuffers gezeigt.

2.1.9 Benutzerverwaltung

Die VisiWinNET Benutzerverwaltung dient zur Vergabe von personenbezogenen Berechtigungen innerhalb der Visualisierungsapplikation. Eingabesteuerelemente können gesperrt, der Zugang zu Informationen über Ausgabesteuerelemente kann verweigert werden.

VisiWinNET unterstützt zwei Verwaltungssysteme für Benutzer:

ebenenorientierte Benutzerverwaltung

Benutzer und Steuerelemente können einer von 999 Ebenen zugeordnet werden. Ist zur Laufzeit die Ebene eines Steuerelementes höher als die des gerade angemeldeten Benutzers, so wird die Bedienung des Steuerelementes abgelehnt.

rechteorientierte Benutzerverwaltung

Den Steuerelementen in der Applikation wird ein Recht zugeordnet, das der Benutzer besitzen muss, um das Steuerelement bedienen zu können (z.B. "Rezept bearbeiten"). Anschließend werden den Benutzern die Rechte zugeteilt, die sie ausführen dürfen.

Dies ist insbesondere in einem Client-Server-System sinnvoll, wenn mehrere Mitarbeiterhierarchien (Gruppen von Mitarbeitern, die unabhängig voneinander in Berechtigungsebenen aufgeteilt sind) im Netzwerk arbeiten. Über den Vergleich der Rechte von Benutzern und Steuerelementen wird über die Freigabe entschieden.

Die Einstellung des Systems erfolgt im Benutzerverwaltungseditor unter dem Menüeintrag Extras→Konfiguration (Einstellung "System")

Die Verwaltung von Benutzern wird über drei Definitionen in der Projektdatenbank vorgenommen:

Benutzergruppe

Fasst Benutzer zu einer Gruppe zusammen und enthält die charakteristischen Parameter für alle enthaltenen Benutzer. Über die Benutzergruppe werden die Bezüge auf die zur Bedienung der Applikation nötigen Rechte verwaltet.

Benutzer

Enthält die Zugangsinformationen (Anmeldename und Passwort) und den Aktivierungsstatus eines Benutzers.

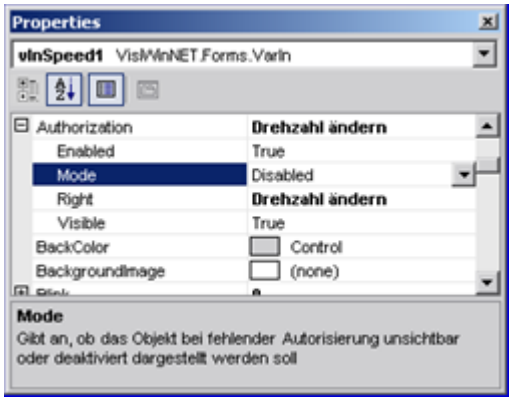
Recht

Rechte definieren die Information, die den Steuerelementen der Applikation zugeordnet werden. Über die "Authorization"-Eigenschaft wird einem Steuerelement ein Recht zugeordnet. Ein angemeldeter Benutzer muss in seiner Benutzergruppendefinition einen Verweis auf dieses Recht haben, damit er das Steurelement bedienen darf.

Rechte werden nur in der rechteorientierten Benutzerverwaltung verwendet. In der ebenenorientierten Benutzerverwaltung erwrtet die "Authorization"-Eigenschaft der Steuerelemente eine Zahl.

Alle drei Definitionstypen können zur Entwicklungszeit über den Benutzerverwaltungseditor projektiert werden.

Alle VisiWinNET-Steuerelemente beinhalten eine 'Authorization'-Eigenschaft. In dieser Eigenschaft wird der Name des Rechtes festgelegt, durch das ein Steuerelement zur Laufzeit bedienbar sein soll. Außerdem wird spezifiziert, wie sich das Steuerelement bei einer fehlenden Berechtigung verhält. Wird die Eigenschaft im Eigenschaftenfenster expandiert, kann in der darunterliegenden 'Mode'-Einstellung zwischen 'disabled' (deaktiviert) und 'Invisible' (unsichtbar) gewählt werden.



Zur Laufzeit stellt VisiWinNET spezielle Dialoge zur Verwaltung der Benutzerinformationen zur Verfügung. Ein typisches Beispiel ist der Anmeldedialog, der als Eingaben den Benutzernamen und ein Passwort verlangt. Über die hier vorgenommene Anmeldung werden die Steuerelemente automatisch je nach vergebener Berechtigung aktiviert oder deaktiviert. Der Aufruf der Benutzerverwaltungsdialoge wird durch die Komponente "UserManagementDialogs" unterstützt.

Das folgende Beispiel zeigt die Arbeitsschritte zur Implementierung einer rechteorientierten Benutzerverwaltung. Beispielhaft wird gezeigt, wie die Verknüpfung der einzelnen Bestandteile auszuführen ist und wie zur Laufzeit Benutzerdaten angepasst werden können. Enthalten sind ein Administrator sowie zwei Benutzer mit unterschiedlichen Rechten. Mithilfe der 'Authorization'-Eigenschaft werden die entsprechenden Freigaben in den Steuerelementen verknüpft.

Außerdem beinhaltet das Beispiel den Aufruf des Anmeldedialoges und des Administrationsdialoges zur Pflege von Benutzerdaten.

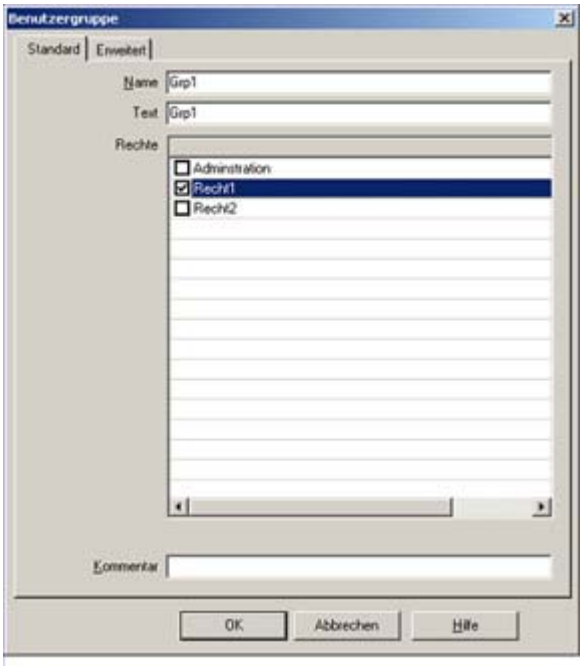
Über einen Doppelklick auf den Knoten "Benutzerverwaltung" im Projektexplorer wird der Tabelleneditor geladen. Hier ist im Menü Extras der Eintrag "Konfiguration" zu wählen. Die Einstellung "System" ist auf "rechteorientiert" zu setzen.

Im Projektexplorer wird der Knoten "Rechte" eingeblendet.

Der "Rechte"-Knoten ist zu markieren. Im Tabelleneditor sind drei Rechte mit den Namen "Recht1", "Recht2" und "Administration" einzutragen.

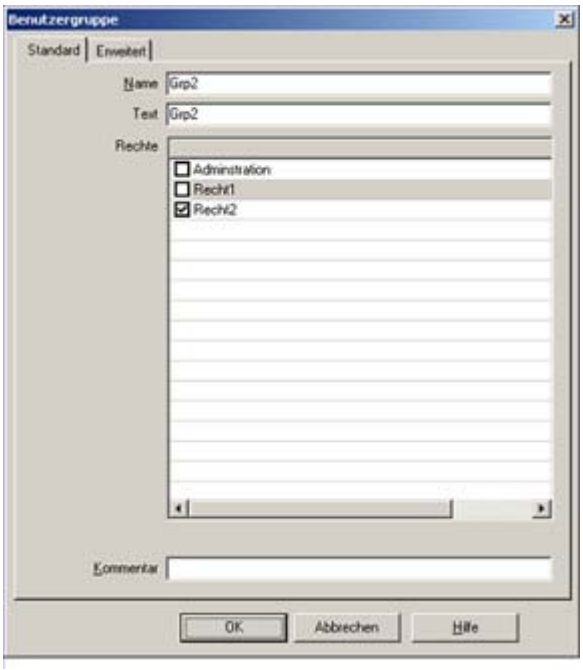
Anschließend ist zu dem Knoten "Benutzergruppen" zu wechseln. Hier sind drei Gruppen anzulegen. Die Parameter sind wie folgt einzustellen:

1. Gruppe



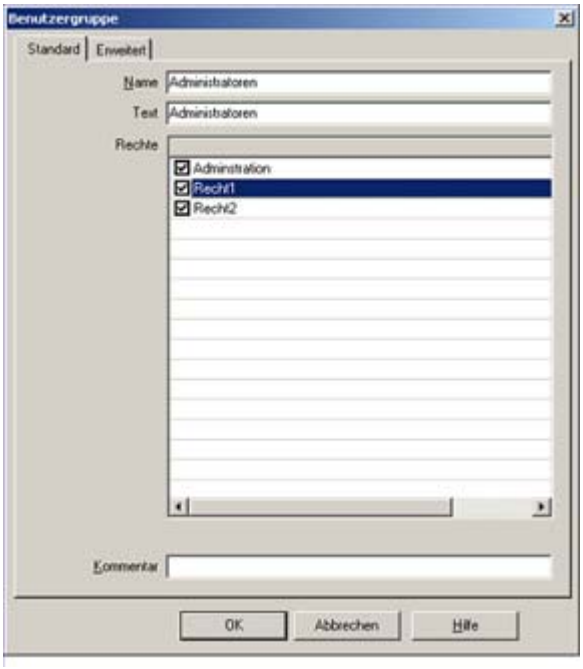
Hier ist der Name anzupassen und die Verknüpfung zu "Recht1" herzustellen.

2. Gruppe



Analog zur ersten Gruppe ist auch hier der Name und die Verknüpfung, wie oben gezeigt, anzupassen.

3.Gruppe



Administratoren dürfen etwas mehr, in diesem Fall alles. Also sind alle Rechte zu verknüpfen.

Die Zuordnung zu Rechten erfolgt über die Benutzergruppe. Das erscheint zwar auf den ersten Blick etwas kompliziert, hilft aber in der realen Welt: Jede Benutzergruppe kann beliebig viele Benutzer aufnehmen. Für alle enthaltenen Benutzer gemeinsam sind die Rechte in der Gruppe einstellbar, also gemeinsam anzupassen.

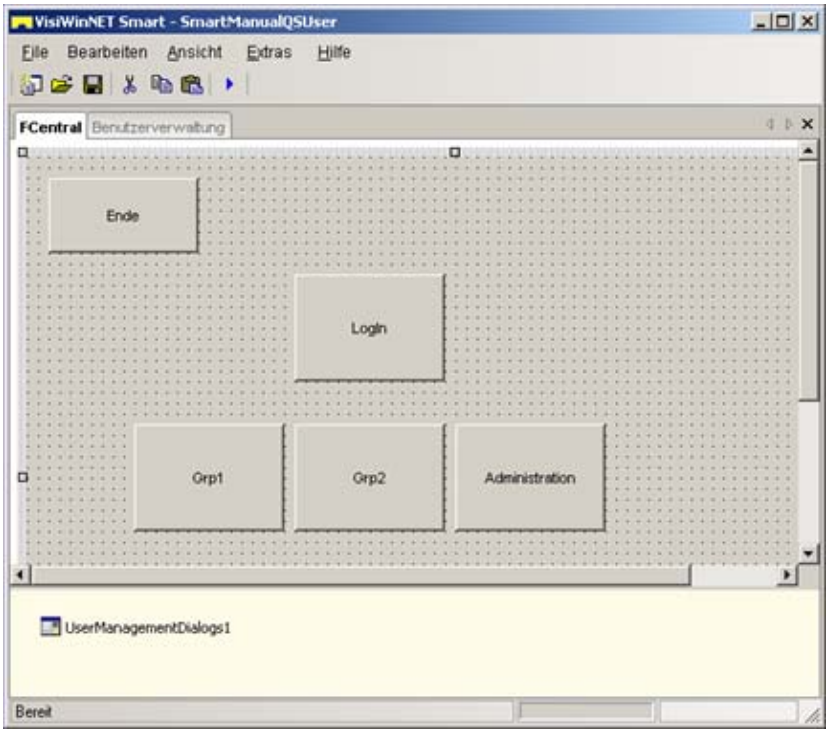
In jeder der drei Benutzergruppen ist im Tabelleneditor ein Benutzer anzulegen. Dabei sind folgende Parameter anzugeben.

Parameter	Wert
Anmeldename	<ein Name>
Passwort	<Ein Passwort das in den Initialeinstellungen mindesten vier Zeichen enthalten muss>
Status	aktiviert

Anmeldename und Passwort werden später als Anmeldeinformationen verwendet. Der Status erlaubt verschiedene Einstellungen, die darüber entscheiden, ob der Benutzer sich anmelden darf und ob er das vom Projekteur vergebene Passwort beim ersten Anmelden ändern muss.

Merken Sie sich die von ihnen vergebenen Anmeldeinformationen für den weiteren Verlauf des Beispiels!

In der Designansicht sind eine 'UserManagementDialogs'-Komponente und vier Schaltflächen vom Typ 'CommandButton' auf einem Formular zu platzieren.



Folgende Eigenschaften sind hier anzupassen:

Für Schaltfläche 1

Text LogOn

Events Das Click-Ereignis ist mit dem Aufruf der 'ShowLogOnDialog'-Funktion der 'UserManagementDialogs'-Komponente zu verknüpfen. Über die ...-Schaltfläche öffnet sich ein Dialog. Hier ist das Click-Ereignis auszuwählen und die "Hinzufügen"-Schaltfläche zu drücken. Auf dem nun sichtbaren Funktionsauswahldialog ist die o.g. Funktion auszuwählen. Nach dem Schließen des Funktionsauswahldialoges ist sicherzustellen, dass die Funktion in der Funktionenliste erscheint und ihr Parameter 'Provider' auf die 'UserManagementDialogs'-Komponente des Formulars verweist.

Für Schaltfläche 2

Text Grp1

Authorization Recht1

Für Schaltfläche 3

Text Grp2

Authorization Recht2

Für Schaltfläche 4

Text Administration

Authorization Administration

Events Das Click-Ereignis ist mit dem Aufruf der 'ShowUserDialog'-Funktion der 'UserManagementDialogs'-Komponente zu verknüpfen (s.o.).

Die Text-Eigenschaft dient hier direkt zur Anzeige der Schaltflächenfunktion.

Die 'Authorization'-Eigenschaft legt jeweils fest, wann die Schaltfläche aktiviert sein soll: Ist ein Benutzer angemeldet und ist seine Benutzergruppe mit dem hier angegebene Recht verknüpft, wird die Schaltfläche aktiviert.

Die Einstellung der 'Events'-Eigenschaft dient zum Aufruf der Dialoge "Benutzer anmelden" und "Benutzer administrieren".

Das Projekt ist zu starten

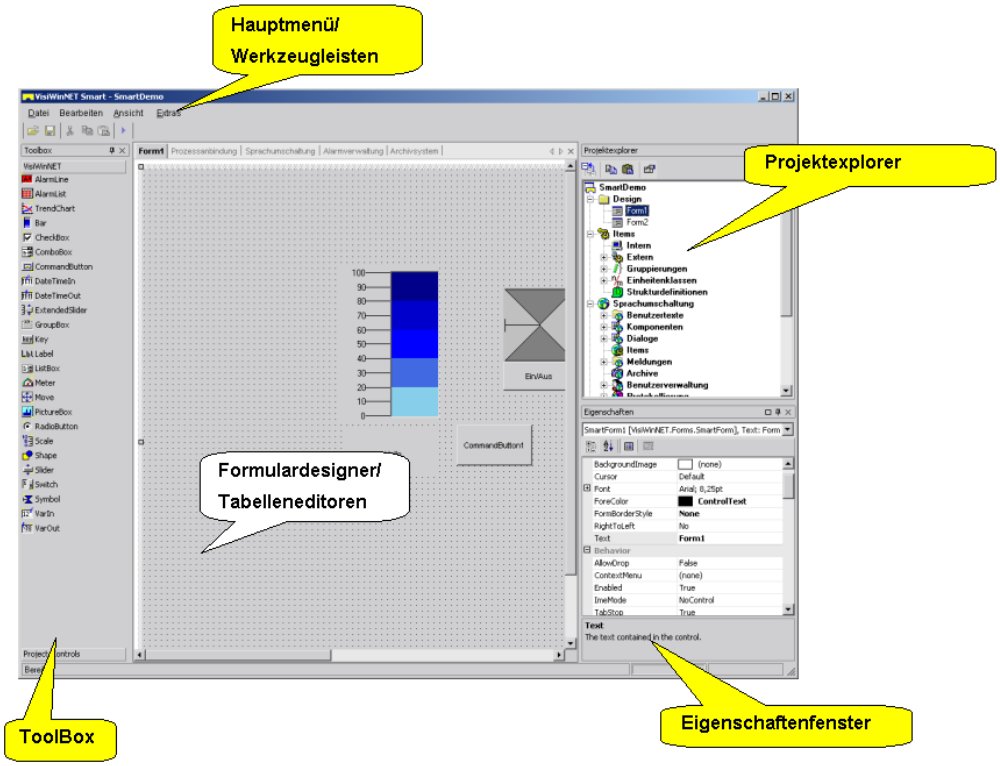
Nach dem Start ist die Schaltfläche "LogOn" freigeschaltet. Hier wurde keine Einstellung in der 'Authorization'-Eigenschaft vorgenommen. Ein Klick auf die Schaltfläche öffnet den Dialog zum Anmelden eines Benutzers.

Die anderen drei Schaltflächen werden durch die Anmeldung der entsprechenden Benutzer freigeschaltet. Die beiden Schaltflächen "Grp1" und "Grp2" sind funktionslos. Sie werden jeweils über die Benutzer aus den Gruppen "Grp1" und "Administration" bzw. "Grp2" und "Administration" freigeschaltet. Werden so parametrierte Schaltflächen dazu verwendet, zu weiteren Formularen zu wechseln, können ganze Applikationsteile in Abhängigkeit vom aktuellen Benutzer freigegeben werden.

Die Schaltfläche "Administration" steht nur Benutzern aus der gleichnamigen Benutzergruppe zur Verfügung. Ein Klick auf die Schaltfläche öffnet den Benutzerverwaltungsdialog. Hier können neue Benutzer angelegt und bestehende Benutzer verwaltet (deaktiviert, gelöscht oder in den Einstellungen geändert) werden.

Die zur Laufzeit geänderten Benutzerdaten werden nicht in der Projektdatenbank gespeichert. Stattdessen erzeigt das System im Projektverzeichnis eine Datei mit der Endung "rtn". So besteht die Möglichkeit,

3 Entwicklungsumgebung



Die Smart-Entwicklungsumgebung stellt auf einen Blick alle wesentlichen Funktionen zur Erstellung eines VisiWinNET-Projektes zur Verfügung.

Projektexplorer

Der Projektexplorer schafft einen Überblick über die Bildschirmansichten und den Inhalt der Projektdatenbank. Über den Projektexplorer lassen sich die Tabelleneditoren der einzelnen Systeme und die Formulare (Bildschirmansichten) des Projektes öffnen.

Formulardesigner

Ein Formulardesigner stellt die Oberfläche einer Bildschirmansicht dar. Eine Applikation besteht üblicherweise aus unterschiedlichen Bildschirmansichten, die die Ansicht und Parametrierung der angeschlossenen Anlagenteile repräsentieren. Über Schaltflächen kann zwischen den Bildschirmansichten gewechselt werden. Jede Bildschirmansicht enthält funktionale Elemente, sogenannte Steuerelemente, die den lesenden oder schreibenden Zugriff auf Prozesswerte und Sonderfunktionen erlauben. Zur Entwicklungszeit werden die Steuerelemente auf der entsprechende Bildsschirmansicht platziert und über das Eigenschaftenfenster parametriert.

Tabelleneditoren

Die Tabelleneditoren repräsentieren die zweite zu projektierende Ebene eines VisiWinNET-Projektes. Jedes Projekt beinhaltet eine Projektdatenbank. Hier werden über die Tabelleneditoren und Funktionen des Projektexplorers zur Entwicklungszeit Definitionen eingetragen und parametriert. Zur Laufzeit steuern diese Definitionen beispielsweise den Datenaustausch zwischen SPS und Applikation. Auch werden hier visualisierungstypische Sonderfunktionen, wie Alarme, Trends, sprachumschaltbare Texte oder Rezepte festgelegt.

Toolbox Die Toolbox enthält die zur Verfügung stehenden Steuerelemente, die auf einem Formulardesigner platziert werden können.

Eigenschaften-fenster Das Eigenschaftenfenster dient zur Parametrierung der Elemente der Visualisierungsoberfläche, also der Steuerelemente und Formulare. Wird ein Steuerelement auf einem Formular selektiert, können seine Eigenschaften im Eigenschaftenfenster betrachtet und geändert werden. Wird im Formular auf eines Stelle geklickt, unter der kein Steuerelement liegt, springt das Eigenschaftenfenster auf die Eigenschaften des Formulars selbst.

Hauptmenü Das Hauptmenü enthält allgemeine Funktionen zum Laden und Verwalten von Projekten. Außerdem können hier die einzelnen Fenster sichtbar geschaltet werden. Für die Bearbeitung von Formularen stehen im Hauptmenü die üblichen Funktionen "Kopieren/Einfügen/Ausschneiden/..." zur Verfügung.

Werkzeugleiste Die Werkzeugleiste beinhaltet Funktionen, die teilweise auch über das Hauptmenü erreichbar sind. Zusätzlich können über die "Layout"-Werkzeugleiste Steuerelemente zueinander ausgerichtet werden.

Schaltflächen der angedockten Fenster

Die an den Rändern der Entwicklungsumgebung verankerten (angedockten) Fenster haben generell in der Titelzeile in der rechten Ecke eine Funktionsleiste mit mehreren Schaltflächen:

Schaltet das Fenster unsichtbar. Über das Menü "Ansicht" kann das Fenster wieder sichtbar geschaltet werden

Diese beiden Symbole beschreiben die Fensterfunktion "automatisch im Hintergrund".

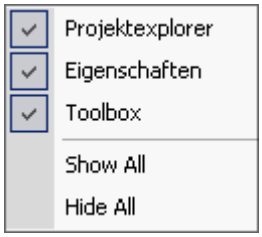
Ist das Symbol zu sehen, wird das Fenster nicht automatisch in den Hintergrund gelegt, wenn es den Fokus verliert.

Ist hingegen das -Symbol sichtbar, wird das Fenster automatisch in der Bildschirmansicht auf Karteireitergröße minimiert, sobald es den Fokus verliert. Dadurch wird die Arbeitsfläche in den anderen Fenstern größer. Wird der Mauszeiger auf den noch sichtbaren Karteireiter geschoben, öffnet sich das Fenster wieder.

Die Schaltfläche maximiert ein Fenster im angedockten Rahmen. In der oben dargestellten Übersichtsgrafik bilden beispielsweise Projektexplorer und Eigenschaftenfenster gemeinsam einen angedockten Rahmen. Das Eigenschaftenfenster kann nun in diesem Rahmen über die -Schaltfläche maximiert werden, so dass der Projektexplorer nicht mehr sichtbar ist. Die Wiederherstellung der ursprünglichen Größe erfolgt über das -Symbol.

Kontextmenü der angedockten Fenster

Wird auf die Titelzeile eines angedockten Fensters geklickt, erscheint folgendes Kontextmenü:



Über dieses Menü können die angedockten Fenster entweder einzeln oder gemeinsam Ein- bzw. ausgeblendet werden.

Verschieben der angedockten Fenster

Die angedockten Fenster können per Drag&Drop verschoben werden:

- Ein angedocktes Fenster wird vom Rand abgelöst, indem es von den Rändern in die Mitte des Bildschirmes weggezogen wird. Danach liegt es in der Entwicklungsumgebung frei verschiebbar im Vordergrund.
- Ein angedocktes Fenster wird an einen anderen Rand verschoben, indem es per Drag&Drop an die gewünschte Kante gezogen wird. Anschließend kann es als angedocktes Fenster in der Größe angepasst werden.
- Ein angedocktes Fenster wird in einen (von einem oder mehreren angedockten Fenstern gebildeten) Rahmen geschoben, indem es per Drag&Drop entweder auf die obere oder auf die untere Kante des Rahmens gezogen wird. Ein Ziehen auf die obere Kante bewirkt, dass das Fenster mit dem bereits dort platzierten Fenster eine Karteireitergruppe bildet. Ein Ziehen auf die untere Kante bewirkt, dass das Fenster unter dem bereits dort platzierten Fenster sichtbar wird.

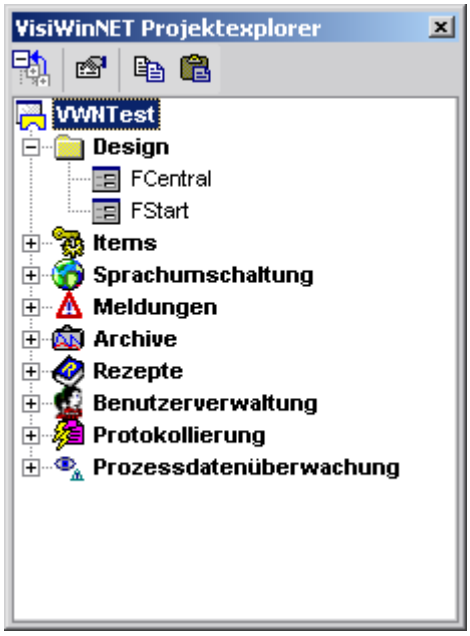
Weitere Bedienungen sind möglich, wenn die Konstellation der Fenster entsprechend ist. Als Hilfestellung dient während des Drag&Drop-Vorganges ein halbtransparenter grauer Rahmen, der die Position des Fensters zeigt, wenn es losgelassen werden würde.

Wechsel zwischen den nicht angedockten Fenster

Formularansichten und Tabelleneditoren werden in der Mitte der Entwicklungsumgebung dargestellt. Sind mehr als ein Fenster geöffnet, kann zwischen ihnen über die Karteireiter gewechselt werden. Sind viele Fenster in diesem Bereich geöffnet, kann es vorkommen, dass nicht mehr alle Karteireiter sichtbar sind. Die Schaltflächen und erlauben einen Bildlauf in den Karteireitern. Die -Schaltfläche schließt das gerade aktive Fenster.

3.1 Projektexplorer

In der Entwicklungsumgebung verwaltet der VisiWinNET-Projektexplorer alle Komponenten des Projektes.



Der VisiWinNET-Projektexplorer gewährleistet den Zugriff auf die Informationen der Projektdatenbank und die Formulare des VisiWinNET-Projektes. Nach dem Start der Entwicklungsumgebung repräsentiert jeder Knoten in der Baumansicht des Projektexplorers ein System. Zusätzlich listet der VisiWinNET-Projektexplorer die Formulare des Projektes auf.

Knoten	Komponente	Funktion
Name	**VisiWin Projekt**	Repräsentiert das VisiWin Projekt durch seinen Namen. • Zusätzlich sind über das Kontextmenü die VisiWinNET-Eigenschaften des Projektes zu erreichen.
Items Meldungen Sprachumschaltung Archive ...	**VisiWinNET-Editoren**	Über die Knoten sind die den einzelnen, unter "Server-komponenten" ausgewählten, Systemen zugeordneten Editoren zu öffnen. Weitere Informationen darüber befinden sich in den entsprechenden Handbüchern.

Design	**Formulare des Projektes**	Über einen Doppelklick auf einen Formulareintrag öffnet sich der entsprechende Formulardesigner. Das Kontextmenü eines Formulareintrages ermöglicht das Löschen oder Umbenennen des Formulars. Im Kontextmenü des Knotens "Formulare" besteht die Möglichkeit, neue Formulare zum Projekt hinzuzufügen.

3.1.1 Bedienung des VisiWinNET Projektexplorers

Die Funktionen des VisiWinNET-Projektexplorers lassen sich in zwei Gruppen teilen:

- generelle Funktionen zum Navigieren im Projektexplorer
- Komponentenfunktionen im Projektexplorer

3.1.1.1 Navigation

Der Zugriff auf die Funktionen einer Komponente wird generell über die Markierung des entsprechenden Knoten in der Baumansicht gesteuert. Ein markierter Knoten wird mit einem blauen Hintergrund im Knotentext dargestellt. Die Navigation erfolgt entweder...

...mit der Maus	Ein Klick mit der Maus markiert einen Knoten in der Baumansicht
...über die Tastatur	Mit den Tasten und kann vom gerade markierten Knoten zum Knoten darunter bzw. darüber gesprungen werden.

Ein Knoten mit einem ⊞-Zeichen enthält weitere Unterknoten. Um Zugriff auf diese Unterknoten zu erlangen, kann der Knoten expandiert werden. Der Expansionsvorgang erfolgt entweder...

...mit der Maus	Ein Klick mit der Maus auf das ⊞-Zeichen expandiert den entsprechenden Knoten in der Baumansicht.
...über die Tastatur	Mit der Taste wird der gerade markierte Knoten expandiert.

Nach diesem Vorgang springt das ⊞-Zeichen um auf das ⊟-Zeichen.

Sind bereits viele Knoten expandiert, wird die Übersicht besser, wenn Knoten wieder zusammengeklappt werden können. Dieser Vorgang erfolgt entweder...

...mit der Maus	Ein Klick mit der Maus auf das ⊟-Zeichen klappt den entsprechenden Knoten in der Baumansicht zusammen.
...über die Tastatur	Mit der Taste wird der gerade markierte Knoten zusammengeklappt.

Nach diesem Vorgang springt das ⊟-Zeichen um auf das ⊞-Zeichen.

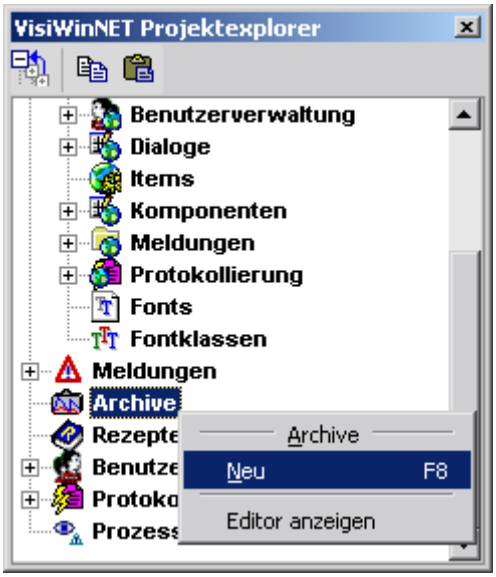
Soll die Baumansicht wieder in den Zustand gebracht werden, der direkt nach dem Start der Entwicklungsumgebung vorlag (Alle Knoten sind zusammengeklappt), so kann dies über die -Schaltfläche in der Werkzeugleiste des Projektexplorers erfolgen.

3.1.1.2 Komponentenfunktionen

Jedes System enthält spezifische Funktionen. Diese Funktionen sind über das Kontextmenü des entsprechenden Knoten im Projektexplorer erreichbar. Das Kontextmenü wird über einen Klick mit der rechten Maustaste auf den entsprechenden Knoten geöffnet.

Viele spezifische Funktionen sind auch durch einen Shortcut erreichbar. Ist für eine Funktion ein Shortcut vorhanden, so wird dieser im Kontextmenü rechts neben dem Funktionsnamen angegeben.

Beispiel:



Der Shortcut zum Anlegen einer neuen Archivdefinition liegt auf der Taste F8. Bei Betätigung dieser Taste wird eine neue Archivdefinition eingefügt.

3.1.2 Schaltflächenbereich

Unter der Titelleiste des Projektexplorers befindet sich ein Bereich mit verschiedenen Schaltflächen:

Dient zum Schließen aller Knoten im Projektexplorer.

Öffnet den Dialog für die Eigenschaften des Projektes.

Kopiert Definitionen aus der VisiWinNET-Projektdatenbank in die Windows®-Zwischenablage. Weiterführende Informationen zu diesem Thema befinden sich im Kapitel "globales Kopieren und Einfügen".

Fügt Definitionen aus der Windows®-Zwischenablage in die VisiWinNET-Projektdatenbank. ein. Weiterführende Informationen zu diesem Thema befinden sich im Kapitel "globales Kopieren und Einfügen".

3.1.2.1 Globales Kopieren und Einfügen

VisiWinNET stellt globale Kopier-/Einfügefunktionen bereit. Diese dienen:

- Zur Übertragung von Definitionen aus einer Projektdatenbank in eine Andere.
- Zum Import von Informationen aus externen Listen.

Bedienung

Im Projektexplorer sind zwei Schaltflächen implementiert, die die gesamte Steuerung der globalen Kopier- und Einfügeoperationen übernehmen

Dient zum Kopieren von Definitionen in die Zwischenablage.

Dient zum Einfügen von Definitionen aus der Zwischenablage in das Projekt.

Über die Zwischenablage steht dem Entwickler die Möglichkeit offen, die kopierten Inhalte durch Texteditoren oder Tabellenkalkulationen weiter zu bearbeiten.

Sowohl der Kopier- als auch der Einfügevorgang sind kontextabhängig, d.h.:

- Beim Kopieren wird der selektierte Inhalt berücksichtigt: Es kann entweder durch die Selektion eines Knotens im Projektexplorer oder durch die Selektion von Definitionen im Tabelleneditor entschieden werden, was in die Zwischenablage kopiert wird.
- Beim Einfügen wird durch die Selektion im Projektexplorer entschieden, welche Definitionen aus der Zwischenablage in das Projekt eingefügt werden.

globales Kopieren

Über die Selektion im Projektexplorer oder im Tabelleneditor wird entschieden, welche Definitionen in die Zwischenablage kopiert werden sollen. Dabei gilt:

- Über den Hauptzugriffsknoten (z.B. "Meldesystem") eines Systems wird der gesamte Inhalt in die Zwischenablage kopiert.
- Über den Zugriffsknoten (z.B. "Meldegruppen") einer Definitionskategorie werden alle Definitionen der Kategorie in die Zwischenablage kopiert. Dabei werden etwaige untergeordnete Definitionen berücksichtigt.
- Wird ein Knoten im Projektexplorer selektiert, der eine Definition repräsentiert, wird diese Definition in die Zwischenablage kopiert. Dabei werden etwaige untergeordnete Definitionen berücksichtigt.
- Auch die Selektion einzelner Definitionen im Tabelleneditor ist erlaubt.

Nach der Selektion wird das Kopieren über die Schaltfläche im Projektexplorer gestartet. Ein Statusfenster gibt Auskunft über den Fortschritt des Kopiervorgangs.

Nach dem Kopiervorgang kann der Inhalt der Zwischenablage in anderen Programmen weiterverarbeitet werden. Bedingung hierfür ist, dass das gewählte Programm TAB-Zeichen NICHT durch Leerzeichen ersetzt. Beispiele hierfür sind "Microsoft® Excel", "Microsoft Word" und der in der "Zubehör"-Programmgruppe liegende "Editor" von Windows® NT/ 2000/ XP. In diese Programme kann der Inhalt der Zwischenablage über die Tastaturkombination "Strg+V" eingefügt werden.

Der Inhalt der Zwischenablage gliedert sich in Definitionsblöcke. Ein Definitionsblock besteht aus drei Elementen:

- In der ersten Zeile steht durch spitze Klammern umgeben der Bezeichner einer Definition.
- In der zweiten Zeile stehen als Tabellenkopf durch TAB-Zeichen getrennt die Feldnamen aus der Projektdatenbank. Diese ermöglichen die Zuordnung der in den folgenden Zeilen angegebenen Werte zu den Parameternamen einer Definition.
- In den folgenden Zeilen stehen durch TAB-Zeichen getrennt die Werte der Definitionsparameter. Dabei wird eine Definition durch eine Zeile in der Zwischenablage dargestellt.

Je nach Selektion werden beim Kopieren ein oder mehrere Definitionsblöcke generiert. Wird z.B. eine Meldegruppe kopiert, werden die Definitionsblöcke "Meldegruppen" und "Meldungen" in der Zwischenablage abgelegt. Beim Kopieren des gesamten Meldesystems über die Selektion des gleichnamigen Knoten im Projektexplorer wird zusätzlich noch der Definitionsblock "Meldeklassen" eingefügt.

Abschließend einige Bemerkungen:

Zusätzlich zu den Parametern einer Definition wird eine Spalte mit der Bezeichnung "[Parent]" generiert. Diese speichert die hierarchische Zuordnung zu übergeordneten Definitionen in Form einer Pfadangabe. Beim späteren Einfügen in ein VisiWinNET-Projekt wird der Inhalt dieser Spalte dazu genutzt, die Schachtelung der einzelnen Definitionen wiederherzustellen.

globales Einfügen

Über die Selektion im Projektexplorer wird entschieden, welche Definitionen aus der Zwischenablage in ein System eingefügt werden sollen. Dabei gilt:

- Der Inhalt der Zwischenablage wird automatisch gefiltert: Es werden nur die Definitionsblöcke aus der Zwischenablage eingefügt, die in dem selektierten System, in der selektierten Kategorie oder in der selektierten Definition einfügbar sind. Als Orientierungshilfe kann die Funktion "Neu" der Editoren dienen. Generell gilt, wenn ein Knoten im Projektexplorer die Funktion "Neu" im Kontextmenü beinhaltet, kann durch die Selektion des Knotens auch eine Definition dieses Typs aus der Zwischenablage eingefügt werden.
- Über den Hauptzugriffsknoten (z.B. "Meldesystem") eines Editors werden alle Definitionen eingefügt, die vom entsprechenden System unterstützt werden. Enthält die Zwischenablage beispielsweise Itemdefinitionen und Meldungsdefinitionen, so werden auch wirklich nur die Meldedefinitionen eingefügt.
- Beim Einfügen aus der Zwischenablage werden die Namensintegritätsregeln der Projektdatenbank überprüft. Definitionen, die schon in der Datenbank vorhanden sind, werden nicht eingefügt. (Bemerkung: in der vorliegenden Version gibt es keine Möglichkeit zum Überschreiben bereits vorhandener Definitionen. Sollen Definitionen in die Projektdatenbank übernommen werden, die bereits vorhanden sind, müssen diese vorher aus dem Projekt gelöscht werden).

- Beim Schreiben der Parameter, die sprachumschaltbare Texte darstellen (z.B. Meldetext der Definition "Meldung" wird keine automatische Erweiterung der Sprachspalten vorgenommen. Sind Texte in der Zwischenablage vorhanden, die für eine Sprache projektiert wurden, die im gewählten Projekt nicht vorhanden ist, werden diese Texte beim Einfügen ignoriert.
- Bezugsdefinitionen (z.B. "Einheitenklassen" als Bezugsdefinition von "Items" in der "Prozessanbindung") werden in der Zwischenablage über den Namen angegeben. Ist die in den Parametern einer Definition angegebene Bezugsdefinition nicht vorhanden, so wird der Bezug als ungültig dargestellt. Bei bestehenden Bezügen ist es also wichtig, dass die Bezugsdefinitionen zuerst in ein Projekt eingefügt werden. Erst anschließend dürfen die Definitionen eingefügt werden, die darauf verweisen.

Beispiele

Im Folgenden zwei Beispiele, die die Bedienung der Kopier-/Einfügeoperationen in der Praxis zeigen.

Projektdaten-übertragung	Der schnellste Weg, Definitionen aus einem alten Projekt in ein Neues zu übernehmen.
Import von Listen	Die Zuordnung von Informationen externer Listen zu Definitionen der Projektdatenbank.

Projektdatenübertragung

Das Beispiel geht davon aus, das ein neues VisiWinNET-Projekt bereits erzeugt und in der Entwicklungsumgebung geladen wurde. Das Projekt, aus dem die Definitionsdaten entnommen werden sollen, heißt hier "PrjSource". Das Projekt, in das die Definitionen der Prozessanbindung eingefügt werden, trägt in der folgenden Beschreibung den Namen "PrjTarget".

Die beiden Projekte sind über den VisiWinNET-Projektmanager zu laden.

Der Knoten "Prozessanbindung" des Projektes "prjSource" ist zu selektieren. Anschließend ist die Schaltfläche im Projektexplorer zu betätigen.

Der gesamte Inhalt der Prozessanbindung aus dem Projekt "PrjSource" wird in die Zwischenablage kopiert. Ein Statusfenster informiert dabei über den Fortschritt des Kopiervorgangs.

Der Knoten "Prozessanbindung" des Projektes "prjTarget" ist zu selektieren. Anschließend ist die Schaltfläche im Projektexplorer zu betätigen.

Der Inhalt der Zwischenablage wird in das Projekt "prjTarget" eingefügt. Ein Statusfenster gibt dabei Aufschluss über den Fortschritt der Aktion.

Im Normalfall läuft der Einfügevorgang in diesem System auf mindestens einen Fehler: Beim Einfügeversuch der Gruppe "Standard" wird ein Fehler ausgelöst, da diese Gruppe in einem neuen Projekt bereits angelegt ist (siehe Handbuch "Prozessanbindung"). Im Statusfenster wird eine entsprechende Fehlermeldung angezeigt.

Import von Listen

Sollen aus anderen Informationsquellen (z.B. Texten, Listen, Tabellen, ...) Informationen in die Projektdatenbank importiert werden, können die globalen Kopier-/Einfügeoperationen verwendet werden. Im folgenden Beispiel soll eine unformatierte Liste von Benutzertexten als Indextexte in die Projektdatenbank übernommen werden. Die Beispielliste ist so aufgebaut, dass deutsche und englische Texte durch ein Semikolon getrennt nebeneinander stehen:

Informationsseite;Infopage

Übersicht;Overview

Meldungen;Messages

.

.

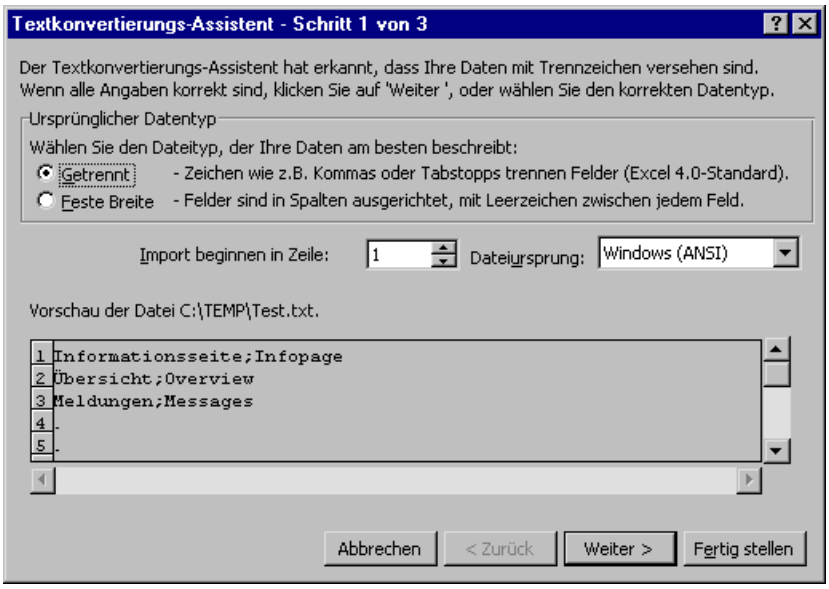
.

Ende;End

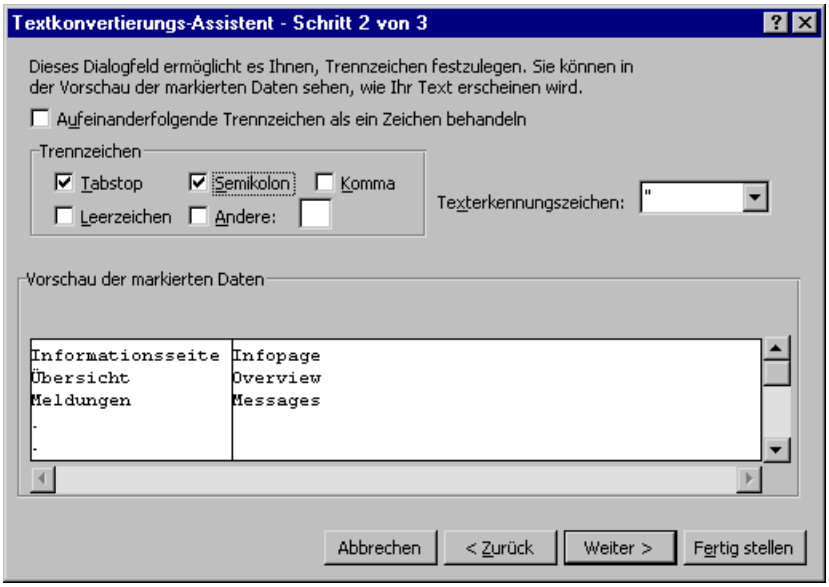
Mithilfe von Microsoft Excel kann die Liste so umformatiert werden, dass sie zum VisiWinNET-Zwischenablageformat kompatibel ist. Folgende Änderungen sind durchzuführen:

Öffnen der Textdatei mit Excel. Im Dateiauswahldialog von Excel ist dabei der Dateitypfilter auf "*.* (alle Dateien)" zu stellen.

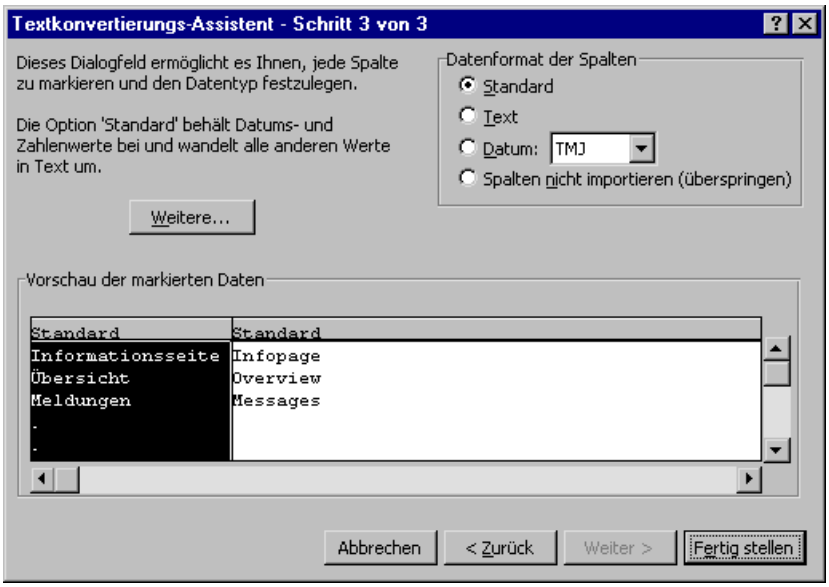
In den folgenden drei Dialogen wird festgelegt, wie die Datei zu öffnen ist.


Textkonvertierungs-Assistent - Schritt 1 von 3
Der Textkonvertierungs-Assistent hat erkannt, dass Ihre Daten mit Trennzeichen versehen sind.
Wenn alle Angaben korrekt sind, klicken Sie auf 'Weiter ', oder wählen Sie den korrekten Datentyp.
Ursprünglicher Datentyp
Wählen Sie den Dateityp, der Ihre Daten am besten beschreibt:
Getrennt - Zeichen wie z.B. Kommas oder Tabstopps trennen Felder (Excel 4.0-Standard).
Feste Breite - Felder sind in Spalten ausgerichtet, mit Leerzeichen zwischen jedem Feld.
Import beginnen in Zeile: 1
Dateiursprung: Windows (ANSI)
Vorschau der Datei C:\TEMP\Test.txt.
1 Informationsseite;Infopage
2 Übersicht;Overview
3 Meldungen;Messages
4 .
5 .
Abbrechen
< Zurück
Weiter >
Fertig stellen

Der erste Dialog kann mit den voreingestellten Festlegungen über die "Weiter"-Schaltfläche bestätigt werden.



In diesem Dialog ist das Trennzeichen "Semikolon" anzuwählen. Dadurch werden die deutschen und englische Texte getrennt als einzelne Felder dargestellt.



Auf diesem Dialog können die Einstellungen auf den Vorgaben belassen werden. Über die Schaltfläche "Fertig stellen" wird der Inhalt der Textdatei in eine Excel-Tabelle eingefügt.

Über die Funktion "Einfügen→Zeilen" wird der Inhalt der Tabelle um eine Zeile nach unten verschoben. Diese Funktion ist zwei mal auszuführen, so dass zwei leere Zeilen entstehen (eine für den Definitionsbezeichner, eine für den Tabellenkopf mit den Parameternamen).

In die erste Zelle der ersten Zeile ist der Definitionsname "<Language.Texts>" einzugeben.

In die zweite Zeile sind folgende Eingaben vorzunehmen:

Zelle1: Text.1031

Zelle2: Text.1033

Zelle 3: Name

Zelle 4: [Parent]

Die Zuordnung der o.g. Eingaben zu den Parametern ist nicht direkt ablesbar. In der Onlinehilfe der einzelnen Systeme wird diese Zuordnung der Parameternamen zu den Datenbankfeldern jedoch durch die Information "Datenbankfeld" ersichtlich.

Eine praktische Hilfe ist auch, wenn bereits vorhandene Definitionen kopiert und in Excel eingefügt werden. Dadurch ist die Zuordnung der Parameternamen zu den selbst vergebenen Parameterwerten ablesbar.

Indextextdefinitionen werden von VisiWinNET über den Namen eindeutig gekennzeichnet. Es wird also vorausgesetzt, dass ein Indextext auch einen Namen hat. Daher müssen in der Spalte "Name" Namen vergeben werden.

Dies kann in Excel auf folgendem Weg sehr einfach getan werden:

- Eingabe des Wertes "Idx1" in der Spalte "Name"der 3. Zeile (Name des Textes "Informationsseite;Infopage").
- Selektion des Feldes
- Bewegen des Mauscursors auf die untere rechte Ecke des Feldes
- Drücken (und halten) der linken Maustaste
- Bewegen des Mauscursors nach unten, bis die Feldselektion neben dem letzten eingegebenen Text (Im Beispiel: Ende;End) steht.
- Loslassen der Maustaste

Excel kopiert in alle jetzt selektierten Felder den Wert des ersten selektierten Feld. Dabei werden die Indizes (die Zahl am Ende des ersten Feldnamen) hochgezählt.

Jetzt also haben alle Texte einen eindeutigen Bezeichner (einen Namen).

Ein VisiWinNET-Projekt ist zu öffnen. Der Knoten "Benutzertexte" im System "Sprachumschaltung" ist zu selektieren. Nach dem Wechsel zu Excel kann der gesamte Inhalt der Tabelle über die beiden Tastaturkombinationen "Strg+A" (für "alles markieren") und "Strg+C" (für "Selektion kopieren") in die Zwischenablage kopiert werden.

Die Entwicklungsumgebung hat die Eigenart, den Inhalt der Zwischenablage beim Start zu überschreiben. Daher muss erst die Entwicklungsumgebung gestartet werden. Danach sind die zu importierenden Daten aus Excel in die Zwischenablage zu kopieren.

Es ist zurück zu VisiWinNET Smart zu wechseln. Über die Schaltfläche ist der Inhalt zu importieren.

Achtung: soll eine Liste aus Excel kopiert werden und enthalten Zellen ein doppeltes Anführungszeichen ("), kann der Inhalt nicht direkt in VisiWinNET eingefügt werden. Dazu sind einige spezielle Arbeitsschritte notwendig:

Die Excel-Mappe ist als Tab-getrennte Textdatei zu speichern

Aus der so erstellten Datei kann mithilfe eines Texteditors der Inhalt in die Windows-Zwischenablage kopiert und in VisiWinNET eingefügt werden.

3.1.3 Spezielle Kontextmenüs

Auf bestimmten Knoten im Projektexplorer sind spezielle Kontextmenüs enthalten, die den Zugriff auf Verwaltungsfunktionen ermöglichen, die nicht spezifisch für die einzelnen Systeme gelten:

Knoten im VisiWinNET-Projektexplorer	Menü	Funktion
Projektknoten	VisiWinNET-Eigenschaften	Öffnet den Dialog zur Festlegung der Projekteigenschaften. In den Projekteigenschaften werden: • die Systeme im Projekt aktiviert oder deaktiviert • die interne Versionshaltung des Projekte verwaltet • die Projektsprache festgelegt.
Knoten "Design"	Formular hinzufügen	Erzeugt ein neues Formular.
Formular-knoten	Designer anzeigen	Öffnet die Designansicht des gewählten Formulars
	Löschen	Löscht den Verweis auf das gewählte Formular aus dem Projekt. Die Formulardatei wird nicht gelöscht.

3.2 Formulardesigner

Der Formulardesigner dient zum Bearbeiten einer Bildschirmansicht. Er wird sichtbar, wenn im Projektexplorer auf einen Formulareintrag doppelt geklickt wird oder wenn aus dem Kontextmenü des Formulareintrag der Eintrag "Designer bearbeiten" gewählt wird. Ist ein Formulardesigner geöffnet, kann er über das -Symbol rechts oben wieder geschlossen werden.

Die Formulardesigner teilen sich mit den Tabelleneditoren der VisiWinNET-Systeme den mittleren Bereich in der Entwicklungsumgebung. Um zwischen den einzelnen geöffneten Fenstern zu wechseln, kann über die Karteireiter oberhalb der geöffneten Fenster navigiert werden.

Die Funktion des Formulardesigners ist eng verknüpft mit zwei weiteren Fenstern der Entwicklungsumgebung:

Toolbox

Die Toolbox enthält die Steuerelemente, die auf dem Formular platziert werden können. Dazu ist zuerst das gewünschte Steuerelement in der Toolbox zu markieren. Anschließend ist auf dem Formular die Stelle durch einen Klick mit der linken Maustaste zu wählen, wo das Steuerelement zu platzieren ist. Wird die Maustaste sofort wieder losgelassen, wird das Steuerelement an dieser Stelle mit einer vorgegebenen Größe platziert. Wird hingegen bei gehaltener Maustaste der Mauszeiger bewegt, bildet die Bewegung auf dem Bildschirm einen Rahmen. Das Loslassen der Maustaste bewirkt, dass das Steuerelement in der Größe des dargestellten Rahmens auf dem Formular erzeugt wird.

Position und Größe von Steuerelementen können nach dem platzieren geändert werden. Wird ein Steuerelement mit einem Mausklick selektiert, wird um das Steuerelement herum ein Rahmen mit Ziehpunkten sichtbar.

Um das Steuerelement zu verschieben, sind folgende Schritte vorzunehmen:

- Selektieren des Steuerelementes durch einen Mausklick
- Drücken (und Halten) der linken Maustaste auf der Steuerelementefläche
- Verschieben des Mauszeigers auf die gewünschte Position
- Loslassen der linken Maustaste

Um die Größe eines Steuerelemente zu ändern, sind folgende Schritte vorzunehmen:

- Selektieren des Steuerelementes durch einen Mausklick
- Drücken (und Halten) der linken Maustaste auf einem Ziehpunkt
- Verschieben des Mauszeigers, bis die gewünschte Größe erreicht ist
- Loslassen der linken Maustaste

Die Ziehpunkte an den Ecken des Rahmens erlauben die gleichzeitige horizontale und vertikale Größenänderung. Die Ziehpunkte an den Rahmenkanten erlauben nur die Größenänderung in jeweils eine Richtung.

Es können auch gleichzeitig mehrere Steuerelemente verschoben oder in der Größe verändert werden. Die Selektion mehrerer Steuerelemente erfolgt über das aufeinanderfolgende Selektieren bei gehaltener STRG-Taste. Typische Hilfsfunktionen zum Ausrichten oder Positionieren von Steuerelementen sind über die "Layout"-Werkzeugleiste aufrufbar.

Eigenschaften-fenster

Nachdem ein Steuerelement auf einem Formular platziert wurde, sind seine Funktionen über das Eigenschaftenfenster parametrierbar. In den Eigenschaften wird festgelegt, "was" das Steuerelement darstellen soll und "wie" der Inhalt aussehen soll. Ein typisches Beispiel für das "was" ist die Festlegung der darzustellenden Prozessgröße über die Eigenschaft "VWItem". Ein Beispiel für das "wie" ist beispielsweise die Einstellung von Farben über "ForeColor" und "BackColor".

Das Eigenschaftenfenster gibt immer die Eigenschaften des selektierten Steuerelementes wieder. Ein Steuerelement wird, wie oben beschrieben, durch einen Mausklick selektiert. Das selektierte Steuerelement wird durch einen Rahmen hervorgehoben.

Ist kein Steuerelement selektiert, werden die Eigenschaften des Formulars selbst angezeigt.

Manche in den Eigenschaften mögliche Einstellungen sind abhängig vom Inhalt der Projektdatenbank. Die VWItem-Eigenschaft enthält einen Dialog zur Auswahl von Prozessgrößen. Diese Auswahl wird aus dem Inhalt der Projektdatenbank generiert. Wenn also keine Prozessgrößen in der Projektdatenbank festgelegt wurde, kann der Dialog nichts anzeigen. Um Items in der Projektdatenbank anzulegen, müssen zuerst im Itemeditor entsprechende Definitionen angelegt werden.

3.3 Tabelleneditoren

Die Tabelleneditoren ermöglichen zusammen mit dem Projektexplorer den Zugriff auf die Definitionen der Projektdatenbank. Hier werden visualisierungsspezifischen Definitionen vorgenommen, die zur Laufzeit den Zugriff auf Prozessdaten erlauben oder Sonderfunktionen, wie Meldungen, Trends oder die Sprachumschaltung beschreiben.

Jedes System von VisiWinNET wird durch einen Tabelleneditor und Einträge im Projektexplorer repräsentiert. Ein Doppelklick auf einen Eintrag im Projektexplorer öffnet den zum Eintrag zugehörigen Tabelleneditor. Die Tabelleneinträge repräsentieren die Definitionen der Projektdatenbank, die unter dem im Projektexplorer markierten Knoten gespeichert sind. Jede Zeile ist eine Definition. Sie enthält die Werte der für die Definition typischen Parameter. Es können neue Definitionen angelegt oder bestehende Definitionen geändert bzw. gelöscht werden.

Ein Teil der Definitionen wird vom Projektexplorer selbst verwaltet. Die vom Projektexplorer verwalteten Definitionen sind üblicherweise Gruppen, in denen sich weitere Definitionen befinden können. Ein typisches Beispiel hierfür sind die Gruppen des Meldesystems, die im Projektexplorer als ein Zweig dargestellt werden. In diesen Gruppen können Untergruppen und Meldungen projektiert werden. Angelegte Untergruppen werden im Projektexplorer dargestellt, weil auch sie wieder Definitionen enthalten können. Meldungen können keine weiteren Definitionen enthalten. Daher werden sie im Tabelleneditor dargestellt.

Nicht alle Knoten im Explorer ermöglichen das Anlegen von Definitionen im Tabelleneditor. Beispielsweise können unter dem Knoten "Archive" keine Trenddefinitionen angelegt werden. Trenddefinitionen können nur in einer übergeordnete Gruppe eine Funktion erfüllen. Diese Gruppe wäre in diesem Fall eine Archivdefinition. Zuerst ist also im Projektexplorer eine Archivdefinition anzulegen. Diese ist anschließend im Projektexplorer zu selektieren. Erst dann wird im Tabelleneditor das Anlegen von Trenddefinitionen möglich.

Teilweise sind bestimmte Gruppen auch gesperrt. Sie sind bereits in einem neuen Projekt angelegt und können nicht gelöscht werden. Der Inhalt dieser Gruppen kann auch nicht im Tabelleneditor erweitert werden. Diese Gruppen enthalten Definitionen, die vom Laufzeitsystem oder von den Steuerelementen erwartet werden. Definitionen in diesen Gruppen können aber parametriert werden.

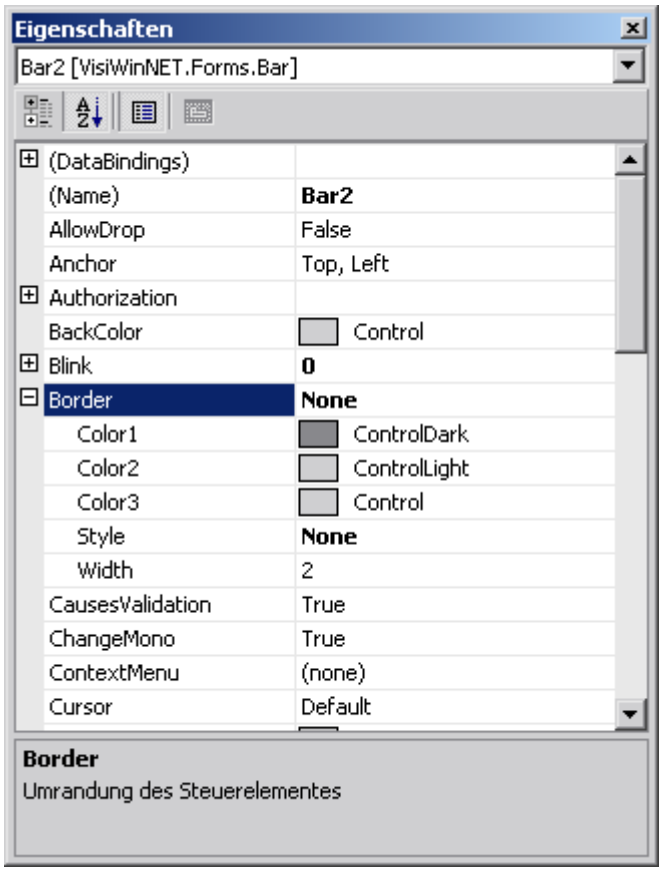
Im wesentlichen haben alle Tabelleneditoren die gleiche Funktionalität:

Definition neu anlegen	Shortcut "F8": Eine neue Zeile wird im Tabelleneditor erzeugt. Üblicherweise wird auch gleich ein automatisch generierter Name für die Definition eingetragen
Definition bearbeiten	Shortcut "F9": Ein Dialog wird geöffnet, der ungeachtet der Tabellenkonfiguration alle Parameter der Definition übersichtlich darstellt.
Definition löschen	Shortcut "Strg+Entf" (nachdem eine oder mehrere Definitionen selektiert worden sind): Die Definition wird aus der Projektdatenbank gelöscht.

Die Tabelleneditoren beinhalten weitestgehend die Möglichkeit der Konfiguration. Über den Menüpunkt Extras→Optionen können einzelne Parameterspalten ein- oder ausgeblendet werden. Auch kann hier die Reihenfolge der Spalten festgelegt werden.

Im Einzelnen werden die Funktionen der Editoren und Definitionen der einzelnen Systeme in den entsprechenden Systemhandbüchern erläutert.

3.4 Eigenschaftenfenster



Das Eigenschaftenfenster dient zur Parametrierung des aktiven Objektes. Nach der Selektion eines Steuerelementes im Formulardesigner z.B. werden im Eigenschaftenfenster die Eigenschaften dieses Steuerelementes aufgelistet. Die Auswahlliste an der oberen Fensterkante zeigt den Namen des in der Formularansicht selektierten Objektes (Hier "Bar2") sowie den entsprechenden Typ des Objektes (hier "[VisiWinNET.Forms.Bar]"). Wird die Auswahlliste ausgeklappt, werden alle auf dem Formular platzierten Steuerelemente, sowie das Formular selbst aufgelistet. Die Auswahl eines Eintrages in der Auswahlliste führt dazu, dass das gewählte Objekt im Formulardesigner selektiert wird und die entsprechenden Eigenschaften im Eigenschaftenfenster angezeigt werden.

Die Liste der zur Verfügung stehenden Eigenschaften ist abhängig vom Typ eines Objektes.

Die linke Spalte zeigt die Namen der Eigenschaften an. In der rechten Spalte wird der aktuelle Wert der Eigenschaft wiedergegeben. Hier können textuell oder über Hilfsdialoge neue Eigenschaftenwerte eingestellt werden. Eigenschaften, die durch ein ⊞-Zeichen angeführt werden, sind ein komplexer Typ und enthalten nach dem expandieren (Klick auf das ⊞-Zeichen) weitere Eigenschaften. In der obigen Darstellung zeigt die Grafik z.B. die Eigenschaften von "Border" an.

Nachdem eine Eigenschaft selektiert wurde, wird in der entsprechenden Wertespalte unter Umständen eine Schaltfläche dargestellt:

- Die -Schaltfläche zeigt an, dass zwischen einer begrenzten Anzahl von möglichen Einstellungen zu wählen ist. Ein Klick auf die Schaltfläche öffnet eine Auswahlliste.
- Die -Schaltfläche zeigt, dass ein Hilfsdialog zur Einstellung der Eigenschaft zur Verfügung steht. Ein Klick auf die Schaltfläche öffnet den eigenschaftenspezifischen Hilfsdialog. Die dort vorgenommenen Einstellungen werden nach Betätigen der OK-Schaltfläche in das Wertefeld der Eigenschaft zurückgeschrieben.

Am unteren Rand des Eigenschaftenfensters ist ein Feld zu sehen, in der der Name und ein kurzer beschreibender Text zur Eigenschaft dargestellt werden. Ist dieser Text nicht ausreichend, kann über die Taste F1 die Hilfeseite der selektierten Eigenschaft aufgerufen werden.

Oberhalb der Eigenschaftenliste sind zwei Schaltflächen angeordnet, die die Darstellung der Eigenschaftenliste steuern:

Unterteilt die Eigenschaften in Kategorien.

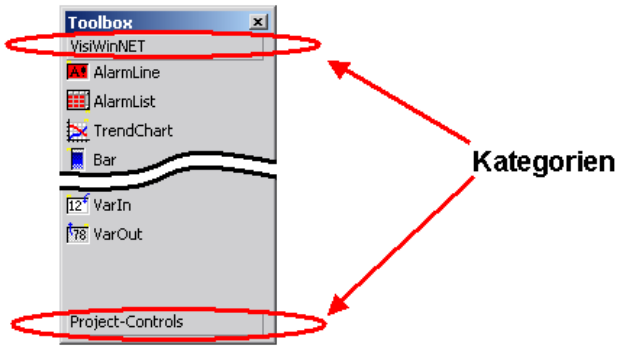
Stellt die Eigenschaften ohne Kategorien in alphabetischer Reihenfolge dar

3.5 Toolbox

Die Toolbox enthält die für die Applikation zur Verfügung stehenden Steuerelementetypen in Form einer Liste. Jeder Steuerelementetyp wird durch ein Symbol und dem Namen des Typs repräsentiert. Um die grafische Oberfläche einer Applikation zu gestalten, sind Steuerelemente auf Formularen zu platzieren. Dazu sind folgende Arbeitsschritte nötig:

- Zuerst ist der gewünschte Steuerelementetyp in der Toolbox zu markieren
- Anschließend ist auf dem Formular die Stelle durch einen Klick mit der linken Maustaste zu wählen, wo das Steuerelement zu platzieren ist. Wird die Maustaste sofort wieder losgelassen, wird das Steuerelement an dieser Stelle mit einer vorgegebenen Größe platziert. Wird hingegen bei gehaltener Maustaste der Mauszeiger bewegt, bildet die Bewegung auf dem Bildschirm einen Rahmen. Das Loslassen der Maustaste bewirkt, dass das Steuerelement mit Größe und Position des dargestellten Rahmens auf dem Formular erzeugt wird.

Die Steuerelemente sind in der Toolbox in der Regel in einzelne Kategorien aufgeteilt.



Ein Klick auf ein Kategoriesymbol schaltet die in der Kategorie enthaltenen Steuerelemente sichtbar. Ist also ein Steuerelement nicht auf Anhieb sichtbar, sollte der Benutzer zuerst einmal die einzelnen Kategorien durchsuchen.

3.6 Menüs

Die Einträge des Hauptmenüs beinhalten allgemeine Funktionen für die Entwicklungsumgebung und den Entwurf von Formularen. Weitere Funktionen von VisiWinNET Smart verstecken sich im Projektexplorer und den Tabelleneditoren der einzelnen Systeme

Datei

Bearbeiten

Ansicht

Zielgerät

Extras

3.6.1 Datei

Projekt Neu	Erzeugt ein neues Projekt
Projekt Öffnen	Öffnet ein VisiWinNET-Projekt
Projekt schließen	Schließt das aktuell in der Entwicklungsumgebung geöffnete Projekt
Speichern	Speichert die Änderungen in den Formularen
Beenden	Beendet die Entwicklungsumgebung

3.6.1.1 Projekt Neu

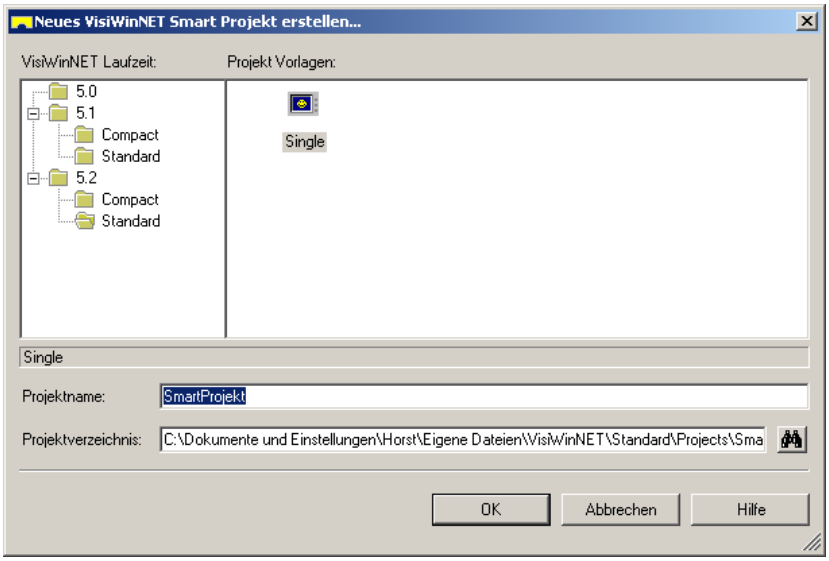
Menü

Datei→Projekt→Neu

Werkzeugleiste

Beschreibung

Öffnet einen Dialog zum Anlegen eines neuen VisiWinNET-Projektes.



In der Baumansicht werden in der obersten Hierarchie die installierten Versionen aufgelistet. Darunter stehen als Zweige die Laufzeittypen zur Auswahl:

Standard Für Standard-PC's: klassische Einzelplatzanwendung, als Enterprise-Variante auch Client-Server-System (mehrere im Netz verteilte Client-Applikationen greifen auf eine Prozessebene zu)

Compact Projekte für Windows-CE oder Embedded Windows XP

Nach Auswahl einer Laufzeit werden in der Vorlagen-Liste die zur Verfügung stehenden Projekttypen angezeigt.

Das Feld "Projektname" benennt das neue Projekt eindeutig..

Auf einem Entwicklungsrechner können niemals zwei Projekte der gleichen Produktvariante mit dem gleichen Namen registriert sein. Daher wird die Eingabe eines bereits bestehenden Projektnamens vom Dialog gemeldet. Es ist jedoch möglich, einen bereits verwendeten Projektnamen zu wählen und ein neues Verzeichnis für das Projekt anzugeben. Dann sind auf dem Rechner zwei gleichnamige Projekte vorhanden. Allerdings kann nur das registrierte (also das neue) Projekt bearbeitet und gestartet werden. Über die "Erweitert"-Karteikarte kann das alte, nicht mehr registrierte Projekt erneut registriert werden.

Das Projektverzeichnis legt den Pfad fest, in den der Projektrahmen gespeichert wird. In dieses Verzeichnis werden später auch alle weiteren Projektdateien gespeichert.

3.6.1.2 Projekt Öffnen

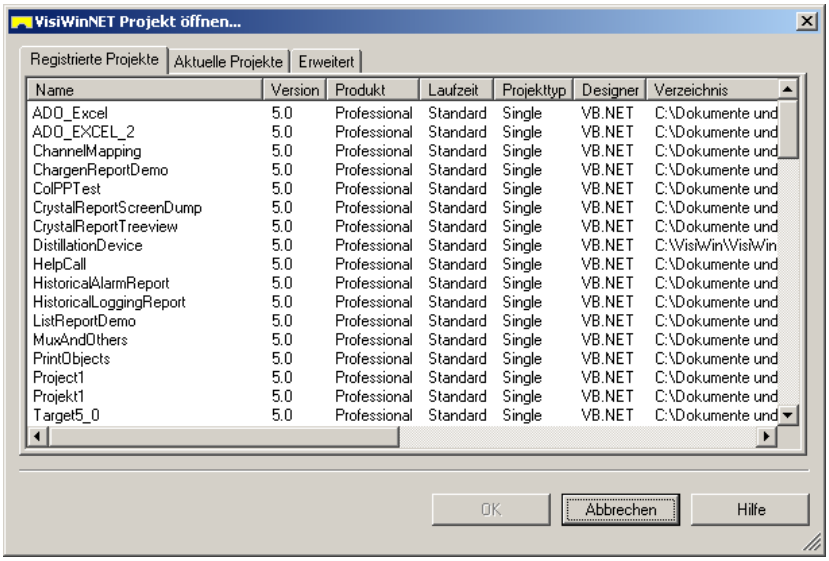
Menü

Datei→Projekt→Öffnen

Werkzeugleiste

Beschreibung

Öffnet ein VisiWinNET- Projekt. Ist bereits ein Projekt geöffnet, wird dieses geschlossen.


VisiWinNET Projekt öffnen...
Registrierte Projekte
Aktuelle Projekte
Erweitert
Name Version Produkt Laufzeit Projekttyp Designer Verzeichnis
ADO_Excel 5.0 Professional Standard Single VB.NET C:\Dokumente und
ADO_EXCEL_2 5.0 Professional Standard Single VB.NET C:\Dokumente und
ChannelMapping 5.0 Professional Standard Single VB.NET C:\Dokumente und
ChargenReportDemo 5.0 Professional Standard Single VB.NET C:\Dokumente und
ColPPTest 5.0 Professional Standard Single VB.NET C:\Dokumente und
CrystalReportScreenDump 5.0 Professional Standard Single VB.NET C:\Dokumente und
CrystalReportTreeview 5.0 Professional Standard Single VB.NET C:\Dokumente und
DistillationDevice 5.0 Professional Standard Single VB.NET C:\VisiWin\VisiWin
HelpCall 5.0 Professional Standard Single VB.NET C:\Dokumente und
HistoricalAlarmReport 5.0 Professional Standard Single VB.NET C:\Dokumente und
HistoricalLoggingReport 5.0 Professional Standard Single VB.NET C:\Dokumente und
ListReportDemo 5.0 Professional Standard Single VB.NET C:\Dokumente und
MuxAndOthers 5.0 Professional Standard Single VB.NET C:\Dokumente und
PrintObjects 5.0 Professional Standard Single VB.NET C:\Dokumente und
Project1 5.0 Professional Standard Single VB.NET C:\Dokumente und
Projekt1 5.0 Professional Standard Single VB.NET C:\Dokumente und
Target5_0 5.0 Professional Standard Single VB.NET C:\Dokumente und
OK
Abbrechen
Hilfe

Der Dialog hat drei Karteikarten:

Registrierte Projekte

Die Karteikarte "Registriert" dient zum Öffnen eines registrierten Projektes. Alle registrierten Projekte werden hier alphabetisch sortiert aufgelistet. Das Öffnen eines der 10 zuletzt bearbeiteten Projekte ist auf der Karteikarte "Aktuell" schneller möglich. Dort sind jedoch nicht alle registrierten Projekte aufgelistet, dies ist insbesondere der Fall wenn

- seit dem letzten Öffnen des Projektes zehn andere Projekte geöffnet wurden, das Projekt also aus der aktuell-Liste herausgefallen ist, oder
- seit dem letzten Öffnen des Projektes eine neue VisiWin-Installation vorgenommen wurde.

Aktuelle Projekte	Die Karteikarte "Aktuell" dient zum schnellen Öffnen eines der 10 zuletzt bearbeiteten Projekte.
Erweitert	Die Karteikarte "Erweitert" dient zum Registrieren eines Projektes Über die Schaltfläche wird ein Dialog geöffnet, der die Auswahl einer VisiWinNET-Projektdatei (Dateiendung *.vwn) ermöglicht. Nach der Auswahl einer VWN-Datei wird das Projekt automatisch registriert. Diese Funktion ist sinnvoll, wenn ein Projekt von einem anderen Entwicklungsrechner kopiert wurde. An dieser Stelle kann auch eine Projektgruppe (Dateiendung *.sln) gesucht und registriert werden.

3.6.1.3 Projekt schließen

Menü: Datei→Projekt→Schließen

Beschreibung: Schließt das in der Entwicklungsumgebung geladene Projekt.

3.6.1.4 Speichern

Menü: Datei→Speichern

Werkzeugleiste:

Beschreibung: Speichert die in der Entwicklungsumgebung geänderten Formulare.

Sind Steuerelemente oder Eigenschaften geändert worden, müssen diese Änderungen gespeichert werden.

Änderungen an der Projektdatenbank müssen nicht gespeichert werden.

3.6.1.5 Beenden

Menü: Datei→Beenden

Beschreibung: Schließt VisiWinNET SMART.

Vor dem Beenden wird der Benutzer unter Umständen dazu aufgefordert, Änderungen an den Formularen des Projektes zu speichern.

3.6.2 Bearbeiten

Ausschneiden Schneidet selektierte Steuerelemente aus dem Formular aus und kopiert sie in die Zwischenablage.

Kopieren Kopiert die selektierten Steuerelemente in die Zwischenablage.

Einfügen Fügt Steuerelemente aus der Zwischenablage in das aktive

Löschen Löscht die selektierten Steuerelemente

3.6.2.1 Ausschneiden

Menü Bearbeiten→Ausschneiden

Werkzeugleiste

Beschreibung Schneidet selektierte Steuerelemente aus dem Formular aus und kopiert sie in die Zwischenablage.

Das selektieren eines Steuerelementes erfolgt durch einen Klick mit der linken Maustaste auf das Steuerelement.

Um mehrere Steuerelemente zu selektieren, sind die Steuerelemente nacheinander mit gedrückter STRG-Taste durch einen Klick mit der linken Maustaste zu markieren.

3.6.2.2 Kopieren

Menü Bearbeiten→Kopieren

Werkzeugleiste

Beschreibung Kopiert die selektierten Steuerelemente in die Zwischenablage.

Das selektieren eines Steuerelementes erfolgt durch einen Klick mit der linken Maustaste auf das Steuerelement.

Um mehrere Steuerelemente zu selektieren, sind die Steuerelemente nacheinander mit gedrückter STRG-Taste durch einen Klick mit der linken Maustaste zu markieren.

3.6.2.3 Einfügen

Menü Bearbeiten→Einfügen

Werkzeugleiste

Beschreibung Fügt die über "Kopieren" oder "Ausschneiden" in die Zwischenablage kopierten Steuerelemente auf dem aktiven Formular ein.

3.6.2.4 Löschen

Menü — Bearbeiten→Löschen

Beschreibung — Löscht die selektierten Steuerelemente vom Formular.

Das selektieren eines Steuerelementes erfolgt durch einen Klick mit der linken Maustaste auf das Steuerelement.

Um mehrere Steuerelemente zu selektieren, sind die Steuerelemente nacheinander mit gedrückter STRG-Taste durch einen Klick mit der linken Maustaste zu markieren.

3.6.3 Ansicht

Projektexplorer — Schaltet den Projektexplorer sichtbar.

Toolbox — Schaltet die Toolbox sichtbar.

Eigenschaften — Schaltet das Eigenschaftenfenster sichtbar.

Startseite — Schaltet die Startseite sichtbar.

Symbolleisten→ Layout — Schaltet die Symbolleiste "Layout" sichtbar.

3.6.3.1 Projektexplorer

Menü — Ansicht→Projektexplorer

Funktion — Schaltet den Projektexplorer sichtbar. Ist das Fenster bereits sichtbar, wird keine Funktion ausgeführt.

3.6.3.2 Toolbox

Menü — Ansicht→Toolbox

Funktion — Schaltet die Toolbox sichtbar. Ist das Fenster bereits sichtbar, wird keine Funktion ausgeführt.

3.6.3.3 Eigenschaften

Menü — Ansicht→Eigenschaften

Funktion — Schaltet das Eigenschaftenfenster sichtbar. Ist das Fenster bereits sichtbar, wird keine Funktion ausgeführt.

3.6.3.4 Startseite

Menü — Ansicht→Startseite

Funktion — Schaltet die Startseite sichtbar. Ist das Fenster bereits sichtbar, wird keine Funktion ausgeführt.

3.6.3.5 Symbolleiste Layout

Menü	Ansicht→Symbolleisten→Layout
Funktion	Schaltet die "Layout"-Werkzeugleiste im Werkzeugleistenbereich sichtbar oder unsichtbar. Die Sichtbarkeit wird im Menü durch ein Kontrollkästchen angezeigt.

3.6.4 Zielgerät

Das Menü "Zielgerät" ist nur sichtbar, wenn ein "Compact"-Projekt in der Entwicklungsumgebung geöffnet ist. Die hier enthaltenen Funktionen steuern die Verbindung zu einem "Windows CE"-Gerät.

Verbindung herstellen	Stellt eine Verbindung zur Übertragung der Projektdaten zu einem "Windows CE"-Gerät her.
Verbindung trennen	Trennt die Verbindung zum "Windows CE"-Gerät

3.6.4.1 Verbindung herstellen

Menü

Zielgerät→Verbindung herstellen

Funktion

Stellt die Verbindung zu einem "Windows CE"-Gerät her.

Auf der Gegenseite wird der VisiWinNET DeviceConnector erwartet.

Verbinden eines Windows CE Gerätes mit VisiWinNET Smart

VisiWinNET Smart-Projekte, die für das VisiWinNET Compact Laufzeitsystem für Windows CE erstellt wurden, können mit VisiWinNET Smart automatisch zum Windows CE Zielsystem übertragen, und dort zum Test gestartet werden.

Der Entwicklungsrechner, auf dem die VisiWinNET Smart Entwicklungsumgebung ausgeführt wird, muss zu diesem Zweck mit dem Windows CE Gerät über ein TCP/IP fähiges Netzwerk verbunden sein.

Vorbereiten der Verbindung:

Die Verbindung zum Zielsystem wird über das Hilfsprogramm "VisiWinNET DeviceConnector" hergestellt. Diese Programm ist Teil der VisiWinNET Smart Installation und muss auf das Windows CE Gerät kopiert werden.

Die entsprechende Datei befindet sich im VisiWin Installationsverzeichnis unter:

<x>:\VisiWin\VisiWinNET\Compact\DeviceConnector\

Die Datei kann z.B. per Windows – Netzwerk über eine Freigabe des Entwicklungsrechners kopiert werden. Unter Windows CE öffnet man dazu das "Start" Menü und wählt den Punkt "Run". Danach gibt man den Rechnernamen in der Form \\MeinPC ein. "MeinPC" steht hier für den Netzwerknamen des Entwicklungsrechners. Nach der Ausführung öffnet sich dann entweder eine Ansicht mit den Freigaben des PCs, oder es wird vorher die Eingabe von Benutzername und Passwort angefordert.

Damit das Programm dauerhaft zur Verfügung steht, ist es zweckmäßig, diese Datei auf der Compact Flash des CE Gerätes abzulegen.

Verbindung herstellen:

Ist das Tool erfolgreich gespeichert, kann es über ein Doppelklick auf dem CE Gerät gestartet werden. Der VisiWinNET DeviceConnector zeigt seine Verbindungsbereitschaft nun mit der folgenden Meldung an:

Warte an "IP-Adresse" auf Verbindung.

"IP-Adresse" steht hier als Platzhalter für die lokale IP-Adresse des Windows CE Gerätes.

Das Gerät ist nun bereit, die Verbindung mit der VisiWinNET Smart Entwicklungsumgebung aufzunehmen. Beim ersten Start des VisiWinNET Projektes in der VisiWinNET Smart Entwicklungsumgebung über die Schaltfläche "Projekt starten", wird ein Dialog eingeblendet, in den die IP-Adresse des Zielgerätes einzugeben ist. Hier muss nun die vom VisiWinNET DeviceConnector angezeigte IP-Adresse eingegeben werden.

Nach dem Betätigen der Schaltfläche "Verbindung herstellen" baut die VisiWinNET Smart Entwicklungsumgebung die Verbindung auf und überträgt alle notwendigen Dateien auf das Zielsystem. Anschließend wird das Projekt automatisch gestartet.

Die Verbindung bleibt so lange erhalten, bis entweder das Projekt in der Entwicklungsumgebung geschlossen oder die Verbindung händisch über den Menüpunkt "Verbindung trennen" im Menü "Zielgerät" der Entwicklungsumgebung beendet wird.

Bekannte Problem:

Die hier beschriebene Verbindung funktioniert recht zuverlässig. Vereinzelt gab es Probleme in Firmennetzwerken, in denen Ports gesperrt sind. Eine direkte Punkt zu Punkt Verbindung zwischen Gerät und PC (z.B. Crosslink Kabel) war bislang immer erfolgreich.

Dauerhafte Installation der Anwendung:

Die VisiWinNET Smart Entwicklungsumgebung kopiert das Projekt in einen Unterordner mit dem aktuellen Projektnamen ins Stammverzeichnis des Windows CE Gerätes. Dieser Speicherort ist bei den meisten Windows CE Geräten im flüchtigen RAM Bereich des Gerätes untergebracht. Das Projekt geht somit beim Abschalten des Gerätes verloren. Soll das Projekt dauerhaft auf dem Gerät bleiben, kann einfach das gesamte Projektverzeichnis auf die Compact Flash Karte kopiert werden. Vorausgesetzt, der OPC Server und das Compact Framework ist auf dem Gerät ordnungsgemäß installiert, kann die Applikation nach einem Neustart einfach aus dem Verzeichnis der Compact Flash wieder gestartet werden.

3.6.4.2 Verbindung trennen

Menü	Zielgerät→Verbindung trennen
Funktion	Trennt die Verbindung zum "Windows CE" Zielgerät

3.6.5 Extras

Optionen Legt allgemeine Optionen für die Entwicklungsumgebung fest

3.6.5.1 Optionen

Dieser Dialog befindet sich zur Zeit in der Entwicklung

3.7 Werkzeugleisten und Schaltflächen

Neben der "Layout"-Werkzeugleiste beinhaltet die Smart-Entwicklungsumgebung folgende immer wieder benötigte Schaltflächen im Werkzeugleistenbereich:

	Öffnet ein VisiWinNET Projekt
	Speichert die bearbeiteten Formulare
	Schneidet die selektierten Steuerelemente aus und kopiert sie in die Zwischenablage
	Kopiert die selektierten Steuerelemente in die Zwischenablage
	Fügt die Steuerelemente aus der Zwischenablage in das aktive Formular ein

3.7.1 Werkzeugleiste "Layout"

Die "Layout_Werkzeugleiste wird über das Menü "Ansicht→Symbolleisten→Layout" sichtbar geschaltet.

Mithilfe der Werkzeugleiste "Layout" können Steuerelemente einzeln oder in Gruppen positioniert oder aneinander ausgerichtet werden. Dazu sind die Steuerelemente bei gedrückter STRG-Taste nacheinander zu markieren. Zuletzt ist das Steuerelement zu markieren, nach dem die anderen Steuerelemente ausgerichtet werden sollen. Die endgültige Position der anderen Steuerelemente hängt von der Position des letzten selektierten Steuerelements ab.

Hinweis Die Ziehpunkte des letzten selektierten Steuerelements werden gefüllt dargestellt, während die Ziehpunkte der restlichen Steuerelemente leer sind. Alle weiteren Größenänderungen oder Ausrichtungen basieren auf dem letzten selektierten Steuerelement.

Die Schaltflächen der "Layout"-Werkzeugleiste beinhalten folgende Funktionen

Schaltfläche	Funktion
	Am Raster ausrichten
	Links ausrichten
	Vertikal zentriert ausrichten
	Rechts ausrichten
	Oben ausrichten
	Horizontal zentriert ausrichten
	Unten ausrichten
	Breite anpassen
	Größe auf Raster anpassen
	Höhe anpassen
	Größe anpassen
	Horizontalen Abstand angleichen

	Horizontalen Abstand vergrößern
	Horizontalen Abstand verkleinern
	Horizontalen Abstand auf 0 setzen
	Vertikalen Abstand angleichen
	Vertikalen Abstand vergrößern
	Vertikalen Abstand verkleinern
	Vertikalen Abstand auf 0 setzen
	Auf die horizontale Mittenposition des Containers verschieben
	Auf die vertikale Mittenposition des Containers verschieben
	In den Vordergrund
	In den Hintergrund